2006-03

Valued information at the right time (VIRT) and the Navy's cooperative engagement capability (CEC) - a win/win proposition

Acevedo, Rafael A.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/2864

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

VALUED INFORMATION AT THE RIGHT TIME (VIRT) AND THE NAVY'S COOPERATIVE ENGAGEMENT CAPABILITY (CEC) – A WIN/WIN PROPOSITION

by

Rafael A. Acevedo

March 2006

Thesis Advisor:	Rick Hayes-Roth
Co-Advisor:	Curtis Blais

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2006	Master's Thesis

4. TITLE AND SUBTITLE: Valued Information at the Right Time (VIRT) and the Navy's Cooperative Engagement Capability (CEC) – A Win/Win Proposition

5. FUNDING NUMBERS

6. AUTHOR(S) LCDR Rafael A. Acevedo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

In this thesis I examine the theory of Valued Information at the Right Time (VIRT) and the benefits its implementation can provide to the Navy's best example of accurate information-sharing, the Cooperative Engagement Capability (CEC). The primary premise of VIRT is that only information which has some value to the user and could impact mission accomplishment should be allowed to flow from a source to the user. If information has little or no value to the individual it is destined for, it must simply be regarded as overhead and should not be sent/received. Using a simple simulation I show in this thesis that VIRT has the potential to provide benefits of orders of magnitude versus a non-VIRT implementation. The Navy's CEC program represents a premier air track data sharing mechanism. It enables ships augmented with this capability and residing on the network to share fire control quality information on the individual parameters of air tracks such as location, course, speed, and altitude. There is a place for VIRT implementation within CEC. Such an implementation can prove beneficial both to CEC as an internal user of information and also as a supplier to external entities of its valuable track information. Finally, I provide a notional VIRT-enabled, product-line architecture for a coalition information-sharing system. If both the concept of VIRT and CEC are to have a place in the future of information-sharing, the issue of providing this information to our coalition partners must be addressed.

14. SUBJECT TERMS
VIRT, Information-sharing, Track Management, Cooperative Engagement Capability (CEC), Global Information Grid (GIG), Reduced Bandwidth, Minimize Resources, SIMKIT, VISKIT, Simulation, Modeling

15. NUMBER OF PAGES 125

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

VALUED INFORMATION AT THE RIGHT TIME (VIRT) AND THE NAVY'S COOPERATIVE ENGAGEMENT CAPABILITY (CEC) – A WIN/WIN PROPOSITION

Rafael A. Acevedo
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March 2006

Author: Rafael A. Acevedo

Approved by: Rick Hayes-Roth
Thesis Advisor

Curtis Blais
Co-Advisor

Dan Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis I examine the theory of Valued Information at the Right Time (VIRT) and the benefits its implementation can provide to the Navy's best example of accurate information-sharing, the Cooperative Engagement Capability (CEC). The primary premise of VIRT is that only information which has some value to the user and could impact mission accomplishment should be allowed to flow from a source to the user. If information has little or no value to the individual it is destined for, it must simply be regarded as overhead and should not be sent/received. Using a simple simulation I show in this thesis that VIRT has the potential to provide benefits of orders of magnitude versus a non-VIRT implementation. The Navy's CEC program represents a premier air track data sharing mechanism. It enables ships augmented with this capability and residing on the network to share fire control quality information on the individual parameters of air tracks such as location, course, speed, and altitude. There is a place for VIRT implementation within CEC. Such an implementation can prove beneficial both to CEC as an internal user of information and also as a supplier to external entities of its valuable track information. Finally, I provide a notional VIRT-enabled product-line architecture for a coalition information-sharing system. If both the concept of VIRT and CEC are to have a place in the future of information-sharing, the issue of providing timely and valuable information to our coalition partners must be addressed.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS

Advanced Coalition Information Distribution System	ACID
Advanced Concept Technology Demonstration	ACTD
Aircraft Carrier (Nuclear)	CVN
Air Marine Operations Center	AMOC
Carrier Strike Group	CSG
Coalition Secure Management Operating System	COSMOS
Combined Enterprise Regional Information Exchange System	CENTRIXS
Commanding Officer	CO
Commercial-Off-The-Shelf	COTS
Common Operational Picture	COP
Common Relevant Operational Picture	CROP
Comprehensive Maritime Awareness	CMA
Condition Monitor	CM
Condition of Interest	COI
Condition Specifier	CS
Cooperative Engagement Capability	CEC
COP Alignment Module	COPAM
Cruiser (Guided Missile)	CG
Department of Defense	DoD
Department of Homeland Security	DHS
Destroyer	DD
Destroyer (Guided Missile)	DDG
Discrete Event Simulation	DES
European Command	EUCOM
Extensible Markup Language	XML
Extremely High Frequency	EHF
Fast Frigate (Guided Missile)	FFG
Federal Aviation Administration	FAA
FORCEnet Services Infrastructure	FSI
Global Information Grid	GIG
Global Positioning System	GPS
Hard Disk Drive	HDD
High Frequency	HF
Identification Friend or Foe	IFF
Information Directory	ID
Information Store	IS
Internet Protocol	IP
Joint Concept Technology Demonstration	JCTD
Joint Mission Design Tool	JMDT
Kilobits per Second	Kbps
Kilobytes per Second	KBps
Littoral Combat Ship	LCS

Megabits per Second	Mbps
Megabytes per Second	MBps
Meteorology	METOC
Mine Warfare	MIW
Military Operations Other Than War	MOOTW
Naval Network Warfare Command	NETWARCOM
Naval Studies Board	NSB
North American Aerospace Defense Command	NORAD
Northern Command	NORTHCOM
Pacific Command	PACOM
Position of Intended Movement	PIM
Private First Class	PFC
Processing Entity	PE
Product-Line Architecture	PLA
Program Executive Office Integrated Warfare Systems	PEO IWS
Program of Record	POR
Query Planner	QP
Query Specifier	QS
Satellite Communications	SATCOM
Secure Internet Protocol Routing	SIPRNET
Services Oriented Architecture	SOA
Single Integrated Air Picture	SIAP
Software Engineering Institute	SEI
Software Product-Line	SPL
Super High Frequency	SHF
Surface Action Group	SAG
Transmission Control Protocol	TCP
Ultra High Frequency	UHF
Undersea Warfare	USW
Valued Information At The Right Time	VIRT

ACKNOWLEDGMENTS

Mr. Dennis Mania and the staff of the Air Marine Operations Center (AMOC) in Riverside, CA. The tour of your facility and your insight into its operation were very valuable.

Mr. Jim Patzer and Mr. Gary Gafke for your support with coordinating receipt of the CEC software.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM DESCRIPTION

1. Information-sharing and National Security

On September 11, 2001 19 hijackers took control of four aircraft and used them to claim the lives of thousands of innocent Americans. The resultant outcry for both justice and answers was unprecedented. This outcry led to the commissioning of a special committee to find answers – answers the American public demanded. Fast forward three years. The commission delivers its report and the results are clear. There is a distinct lack of information-sharing within Governmental agencies. More over, this lack of information-sharing directly contributed to the events of September 11th. The following excerpt from the 9/11 Commission Report demonstrates the depth of the communication and information-sharing problem:

> On 9/11, the defense of U.S. air space depended on close interaction between two federal agencies: the Federal Aviation Administration (FAA) and North American Aerospace Defense Command (NORAD). Existing protocols on 9/11 were unsuited in every respect for an attack in which hijacked planes were used as weapons…What ensued was a hurried attempt to improvise a defense by civilians who had never handled a hijacked aircraft that attempted to disappear, and by a military unprepared for the transformation of commercial aircraft into weapons of mass destruction. A shootdown authorization was not communicated to the NORAD air defense sector until 28 minutes after United 93 had crashed in Pennsylvania. Planes were scrambled, but ineffectively, as they did not know where to go or what targets they were to intercept. And once the shootdown order was given, it was not communicated to the pilots. In short, while leaders in Washington believed that the fighters circling above them had been instructed to "take out" hostile aircraft, the only orders actually conveyed to the pilots were to "ID type and tail." (9/11 Commission, Executive Summary)

Drastic measures need to be taken in order to ensure that we are prepared to meet the threats of the future. Knowledge is power. That phrase has never been truer than it is today. From that we can infer that *shared*

knowledge then is *ultimate* power. Our organizations need to get information quickly to those who need it without overwhelming them with useless data. The need to separate the "wheat from the chaff" is paramount to avoid overloading both our systems and the personnel who operate them. The 9/11 Commission was clear in their recommendations:

> The U.S. government has access to a vast amount of information. But it has a weak system for processing and using what it has. The system of "need to know" should be replaced by a system of "need to share." (9/11 Commission Report)

Recognizing the fundamental importance of information-sharing, the Department of Defense (DoD) has begun work on an information technology (IT) enabled, service oriented architecture aimed at getting the information to those warfighters who require it. In theory it will act as a one-stop, all inclusive, information shopping place similar to the civilian Internet. On the surface this seems to be a great answer to an information-sharing problem. And in fact, I would agree that it is a crucial piece, the backbone in essence of a solution. What the Global Information Grid (GIG), as currently envisioned, will not do is prevent information and system resource overload. The GIG's primary purpose is to allow those that need information to have secure access to that information. It will likely do this without any regard to exactly what the user truly cares about. It will be at best a *Smart-Pull*[1] system (which we will discuss at length a bit later in this thesis). This leads us to the second problem this thesis attempts to address, the ever-increasing demand for bandwidth – a significant and currently unavoidable prerequisite for information-sharing.

2. Bandwidth

Bandwidth is not an unlimited resource. As we move forward to the future and Net Centric Operations and Warfare (NCOW), both our systems and our personnel will become significantly constrained by bandwidth. In fact this is arguably already becoming so. Growing up I often heard my parents tout the apocryphal money tree that must be replenishing itself daily as I made my next

[1] Smart-Pull is a reference to the status quo, non-VIRT process with which information is generally shared. The user's system *pulls* information from the available information sources.

request for a particular store item. The analogy of unlimited capacity also characterizes naïve concepts of bandwidth. In fact, bandwidth resources are limited and can't be ignored. The Navy already understands how critical and constrained resources should be managed, as is illustrated in the case of fuel. In the Surface Navy when a ship transits somewhere the first thing the crew does is to make sure there is sufficient fuel on hand to make the trip. If the transit requires additional speed, the consumption rate is checked against available fuel stores prior to bringing another engine on line to ensure the ship still has enough fuel to reach the destination. Failure to do so could leave the ship and her crew stranded on the deep blue.

However, in our rush to get new capabilities to our ships, capabilities which often require a significant demand for additional bandwidth, we seem to have forgotten that we need to increase capacity if we are going to increase demand. Our bandwidth capacity has not increased at a rate anywhere close to the actual demand we now place on the system. Communications bandwidth, while the most commonly understood instance, is not the only instance of bandwidth. There are actually two completely different, yet equally important, types of bandwidth that must be accounted for, communications <u>and</u> personnel bandwidth.

a. Communications Bandwidth

When most individuals think of bandwidth, communications bandwidth is generally what they are referring to. Communications bandwidth can be looked at in a couple of different ways but for our purposes communications bandwidth is understood to be the capacity required (or available, depending on the circumstance) of the pathway, or "pipe," that is used for conveying bits. Over the last 11 years of my naval career I have seen the available information systems multiply ten fold. This is primarily true of TCP/IP-based communications systems. The problem is that the capacity of the communications pipe has not increased at the same rate as the development and implementation of software applications that tax this pipe. There are several good examples of just how dependent we are on connectivity (TCP/IP in

particular) these days. This is particularly true of our smaller ships. An Arleigh Burke destroyer (DDG 51 Class), one of the Navy's newest classes of warships, is still limited to 64 kbps.[2] Additionally, this is not a consistent pipe as there are numerous blind zones that can render the system inoperable.

As limited as the available bandwidth is, the demands on the system have significantly increased. Our administration systems now require online updating. Supply departments must check stock of available parts and other equipment through a database accessible only via TCP/IP connections. There is a system that enables technicians to "reach back" and receive technical assistance with emergent problems – all done via the INMARSAT system. The ship's doctor or corpsman must now access a system off ship in order to get the latest and greatest information on current illnesses and also to collaborate on treatment for patients. And let's not forget the morale uses for the systems such as surfing the Internet; as well as personnel professional advancement via online coursework. The Navy is placing a premium on secondary education and many of our enlisted sailors are trying to take advantage of the Internet to get this education. Because Navy personnel are routinely at sea, those sailors rely on our connectivity to enhance their development. All of these uses combine to severely tax the system. Those systems mentioned here are only the unclassified consumers of bandwidth. There are also numerous applications that run on the classified side that require large amounts of bandwidth as well.

Figure 1 below shows the total bandwidth available to afloat forces in 2003. Approximately 192 Mb/s was available. While 192 Mb/s may seem like

[2] There are two possible systems that provide TCP/IP connectivity on a destroyer – single and dual INMARSAT systems. Each INMARSAT is capable of delivering 64kbps. On a ship with two INMARSAT antennas one would expect that the INMARSATs would be additive and provide double the bandwidth. In reality this is not the case as blind zones and blocked courses generally provide for at most one connected INMARSAT system at a time. http://www.chips.navy.mil/archives/03_winter/webpages/pacific.htm.

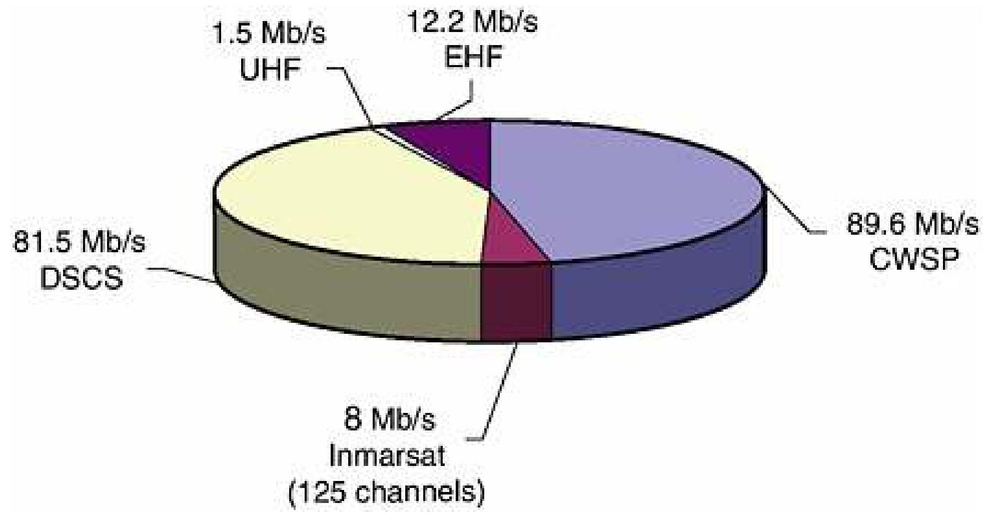

Figure 1. Total estimated available fleet bandwidth for spring 2003. (From, PEO C4S, Spring 2003

a large capacity, remember that this number is what was available for the entire fleet. To put it in perspective, during Operation Iraqi Freedom in the spring of 2003, Combined Joint Forces used over 750 Mb/s continuously. (Moseley, 2003) Our bandwidth demand will only continue to grow over the next several years. The Naval Network Warfare Command (NETWARCOM) estimates that a CVN will require 25 Mb/s by the year 2011. That is over three times the current available bandwidth of 8 Mb/s. (NSB, 2005) Bandwidth availability to surface forces is being increased as shown in Figure 2. However, the projected capacity versus anticipated requirements indicates a significant shortfall. The following excerpt is taken from the report by the Naval Studies Board as they examined the future requirements for space-based communications:

> While the Navy is correct in projecting a general trend of bandwidth growth, the committee believes that the exponential growth in capability- and platform-generated data cause the current naval bandwidth projections to be severely underestimated. Further, the committee believes that the reliance of warfighting capability on satellite communications will necessitate new requirements…The tactical and mobile user will require high-availability, high-bandwidth, assured communications links worldwide. (NSB, 2005)

Platform	Wideband Capacity (Mb/s)		
	FY00 (Estimated)	FY03 (Actual)	FY07 (Projected)
Command ship (LCC)	2.048	3.072	10.496
Aircraft carrier, nuclear-powered (CV/CVN)	2.048	3.072	8.448
Amphibious assault ship (LHD/LHA)	2.048	2.304	8.448
Dock landing ship/amphibious transport dock (LSD/LPD)	0.064	0.064	3.328
Guided cruiser (CG)	0.064	0.384	3.328
Guided missile destroyer (DDG)	0.064	0.128	3.328
Destroyer/guided missile frigate (DD/FFG)	0.064	0.064	3.328
Fast combat support ship (AE/AO/AF)	0.064	0.512	0.512
Attack submarine (SSN)	0.032	0.064	0.512
Guided missile attack submarine (SSGN)	NA	NA	0.768
NOTE: NA = wideband capability not available to platform			

Figure 2. Fleet Bandwidth Availability projected through 2007. (NETWARCOM, 2003)

The Navy and Department of Defense must take a new look at how bandwidth is viewed. Finding ways to reduce bandwidth demands is as important as finding ways to increase available bandwidth.

b. Personnel Bandwidth

Personnel bandwidth is rarely considered when determining information flow limitations. The “Human Factor” is one concept that seems to be missing in most of the point papers and visions of the future. As we increase our information-sharing we put a significant load on our personnel. The only thing certain about warfare in the future is that nothing will be certain. There is a dynamic complexity involved in information-sharing that only adds to the difficulty for the men and women operating or using these systems. Constant change requires operators to process information given them more quickly and efficiently.

In other words, they need to get the highpoints of the information that is relayed while discarding those not needed.

The Navy's focus on minimal manning in future ships will exacerbate information processing challenges faced by personnel. The Navy's newest class of ships, the Littoral Combat Ship (LCS), for example, is one such vessel that will have a minimum crew. Including a mission package the crew will number no more than 125 people.[3] Additionally, the ship will contain the most comprehensive suite of information systems ever placed aboard a naval vessel. This means that fewer people will have to do and process more. Reducing personnel-related costs and increasing information flow would seem to be good things. There will, however, be difficulties that result as these two concepts/policies are implemented. VIRT will help mitigate these difficulties by reducing personnel bandwidth required.

3. Information-sharing with our Coalition Partners

The day of open-ocean warfare amongst superpowers has long since passed. In its place has come Military Operations Other Than War (MOOTW).[4] These operations include humanitarian assistance; disaster relief, terrorist response, hostage rescue, and a host of non-traditional military roles. One of the greatest differences between past and present military functions is the now overwhelming interdependence among global communities. Rather than one nation versus one nation, the focus has shifted considerably over the last fifteen years to a multi-national coalition environment. The problem, however, is that although the military strategy and tactics may have changed, the corresponding infrastructure to support it has not.

[3] The Littoral Combat Ship will have core crew not to exceed 50 people but will have interchangeable mission packages. These mission packages will be people and equipment specifically embarked for a particular mission area such as Mine Warfare (MIW) and Undersea Warfare (USW). With a full complement of core and mission package onboard, crew size will remain limited to 110 personnel. http://www.nps.edu/Research/HCS/Docs/Douangaphaivong_thesis.pdf.

[4] MOOTW is the general term used to describe operations that do not fall underneath the standard umbrella definition of warfare. These include peacemaking and peacekeeping operations as well as non-combatant evacuation operations (NEO). Foreign nation support is also another area of non-traditional operations supported by the United States Military. It is detailed in DoD Directive 3000.05, Military Support for Stability, Security, Transition, and Reconstruction (SSTR) Operations.

We operate with more countries now than ever in the past. Multi-national operations can now mean over 40 different countries are participating. The war in Iraq, for example, includes more than 15 coalition members. Although we have increased our operations with these coalition members, we have not made it a priority to be able to communicate with all of them. In fact, there are many countries with whom we can communicate only through bridge-to-bridge radio. Information-sharing with some of these countries is currently not possible.

It is unlikely that the GIG as currently conceived will enable a connection or service to allow dissemination of information with our allies. This is primarily due to the fact that the Global Information Grid will support all services and other U.S. government agencies. It will contain sensitive and classified data. This will preclude access by the majority of our coalition partners. There are, however, developments currently ongoing to enable coalition information-sharing. One such system is the Coalition Secure Management Operating System (COSMOS). COSMOS is an Advanced Concept and Technology Demonstration (ACTD) that is addressing the issue of information-sharing and interoperability with our coalition partners. COSMOS represents a definite improvement over the current state of information-sharing with our allies. COSMOS does not, however, address the continuing challenge of getting only what is important to the people who need it. The dimension and functionality added by that simple thought is what sets a VIRT-enabled architecture apart from COSMOS and other information-sharing technologies.

Again, the focus here is the development of a product-line, VIRT-enabled architecture that will provide the foundation for not just a single Joint/Coalition system, but potentially many future systems; all of them using the same foundation.

B. THE GOAL

Now we have come to the reason for my work in this thesis. What do I hope to accomplish – that is – what is my goal? The goal of this document is threefold:

First, demonstrate through analysis that a VIRT architecture can prove useful to the Navy and in particular the Cooperative Engagement Capability (CEC) program. As a Surface Warfare Officer I am excited about the prospects and potential that VIRT holds for not just my Warfare Community but the entire Navy. The Cooperative Engagement Capability is the closest thing the Navy has to a truly effective distributed information-sharing network. It only makes sense that as we strive to look to the future we should start with a good product and make it better. CEC can be made better with VIRT.

Secondly show, through a simple simulation, that the Smart-Push process design methodology provides a significant, measurable advantage over the traditional, non-VIRT, Smart-Pull implementation. The Hayes-Roth paper shows a benefit to an operator of five orders of magnitude over a non-VIRT process. I will create an initial simulation model of the node interaction dynamics needed to demonstrate this advantage and to serve as a starting point for follow-on, more detailed analyses.

And lastly, address the shortfall of information-sharing between the United States and its coalition partners. Additionally, I provide a product-line architecture that uses VIRT as a backbone to show the potential for future information-sharing in a joint and multi-national environment. This thesis uses as its primary premise the VIRT model as illustrated in both *Model-based Communications Networks and VIRT*, and *Two Theories of Process Design for Information Superiority: Smart-Pull vs. Smart-Push*, both by Rick Hayes-Roth.

C. THESIS ORGANIZATION

Chapter I has bounded the problem and established specific goals for this thesis. The remaining chapters are focused as follows:

- Chapter II answers the question "What is VIRT?" It sets the foundation of the two Theories of Information flow as set forth by Rick Hayes-Roth.
- Chapter III explains exactly what the Navy's Cooperative Engagement Capability (CEC) program is, how it came about, and

why it is the logical choice for a future in VIRT. Additionally, I introduce the Air Marine Operations Center (AMOC). AMOC is part of the Department of Homeland Security (DHS) and is the best civilian example of a comprehensive information-sharing system.

- Chapter IV introduces the tools used in the simulation models developed and included in this thesis. It also provides a brief explanation of some basic simulation and modeling concepts and sets the framework to allow the reader to understand the results provided in Chapter V.
- Chapter V presents experimental and analytical results. It provides the results and analysis of the two simulation models that were developed to show the Theory 1 and Theory 2 implementations.
- Chapter VI addresses the current state of information sharing amongst coalition members and the United States. Additionally, Appendix A presents a notional product-line architecture for a VIRT-enabled coalition information-sharing system called the Advanced Coalition Information Distribution System (ACID).
- Chapter VII provides the impetus for future research. There are several areas that could be enhanced with additional research and implementation.

II. VALUED INFORMATION AT THE RIGHT TIME (VIRT)

A. TWO THEORIES

We've discussed the problem and I have claimed that VIRT is the answer. Well what exactly is VIRT? VIRT is a concept introduced by Professor Rick Hayes-Roth (RHR) of the Unites States Naval Postgraduate School. It stands for Valued Information at the Right Time. The premise of VIRT is simple: if you get the information to the operators who need it, when they need it, you increase the chances of mission success. (RHR, Model-based Comm Nets). VIRT is not a new system or an Information Technology. It is, however, a radical and unique approach to increase productive information flow and sharing. It is an architectural design that provides the building blocks to enable future information flow. There are in essence two theories to information flow, the traditional way and the VIRT-enabled way.

1. Theory 1 – Smart-Pull

There is always more than one way to accomplish a goal. Sometimes one is superior to the other depending on the conditions, but there is always more than one way. Let's consider for a moment the two alternative design approaches to information flow as espoused by Hayes-Roth in his paper *Two Theories of Process Design for Information Superiority: Smart-Pull vs. Smart-Push*. Smart-Pull is a non-VIRT process. It focuses on frequent user interaction with a multitude of information sources in order to retrieve information.

> Theory 1: Describe all information available using some type of meta-data description. Give each processing entity good search tools. Have each entity seek and acquire whatever information it needs, when and as needed. (Hayes-Roth, 2006)

Theory 1, Smart-Pull, is in essence the status-quo of information-sharing. It's what we have been doing for years. It works as follows: A specific organization decides what information it needs. It is then connected to a network containing one or more independent information sources and continuously (or near continuously) requests (or Pulls) information as required. The system

responds with all available information matching the request. The system responds regardless of whether or not the information has changed, and also without thought to how it may impact the user. This means the user receives the same information again and again and is forced to determine if and how that information has changed. Let’s take a closer look at Theory 1 and the Hayes-Roth model shown in Figure 3.

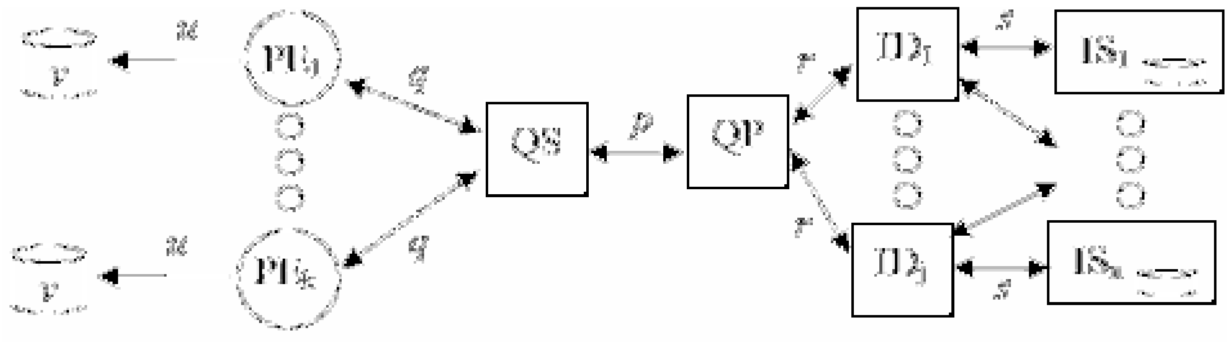

Figure 3. A value chain of processing entities PE_i producing products *v* as a result of specifying queries and planning and executing those queries through information directories to various information sources. (From: Hayes-Roth, 2006)

The entire premise of Theory 1 is to take queries initiated from the individual units, process them against whatever sources are available and provide that information back to the requesting site. The requesting site then takes this information and extracts some value (*v*) from the information and uses it accordingly. The easiest way to understand the Theory 1 model presented in Figure 3 is to relate it to a fictional – yet possible – realistic Theory 1 implementation. Let’s assume a Carrier Strike Group (CSG) is transiting a choke point such as the Strait of Malacca. The strike group is composed of one aircraft carrier (CVN) with embarked air wing (CVW), one Spruance class destroyer (DD), two Ticonderoga class cruisers (CG), and one Arleigh Burke class destroyer (DDG). The resulting composition is shown in Figure 4.

One of the first things to note is that the enabling architecture portrayed in this example assumes the GIG is in place. While I do believe the GIG will be inadequate as currently envisioned, nevertheless, it represents a perfect example

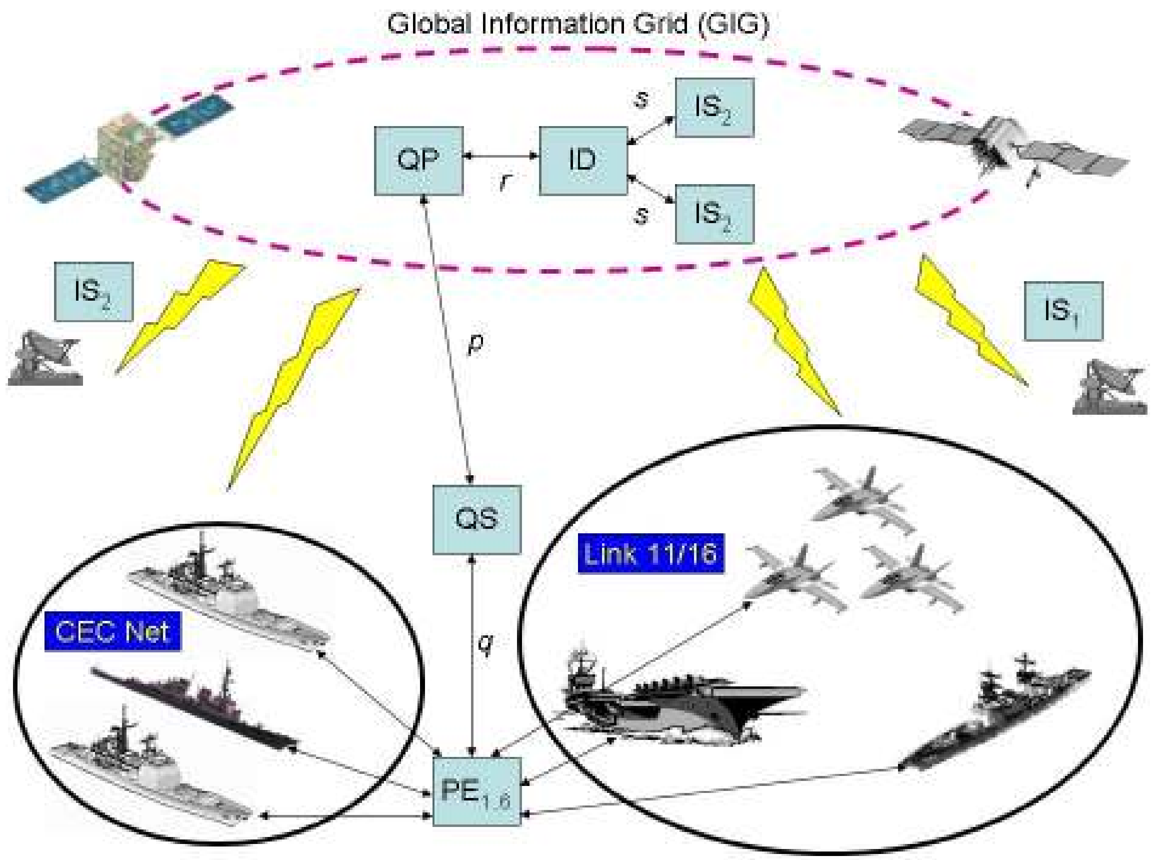

Figure 4. Real World Theory 1 Implementation

of a Smart-Pull focus. The GIG will provide a central network or hub architecture upon which multiple services will run.

The Processing Entities (PE_1 through PE_6 as shown) represent the individual ships and aircraft who actually desire the information (the Carrier Strike Group in our instantiation of Theory 1). The PEs utilize whatever processing resources are required to generate the query. Under Theory 1 the PEs formulate query requests (*q*) through the Query Specifier (QS). The Query Specifier works with the Processing Entity. Its job is to take the requests submitted by the PEs and forward them to the Query Processor (QP) for action. In addition, it provides the information requested back to the PEs so the information can be displayed/used.

The location of the Query Specifier as depicted between the networks and below the GIG is no accident. The primary reason for this is because the QS can ride on either the PE or the backbone of the GIG. It will be a tradeoff between a decision to use the bandwidth/resources of the PE or that of the GIG.

In our case the CVN is the PE. It submits a request for two particular pieces of information to the "system." These include weather and anticipated traffic in and around the strait. That request (*q*) goes to the Query Specifier which communicates with the Query Planner (QP), located on the GIG, to create a Query Plan (*p*) to retrieve the information.

The Query Planner must be located on the GIG because it is assumed that the Query Planner will be receiving query requests from multiple PEs. This is due to the fact that our fictional CSG is not the only one in the Navy. There are currently 12 CSGs and nearly 200 ships. And this is just the Navy side. There will also be requests from other services (such as the USAF, USMC, and USCG). Additionally, each PE will be submitting multiple requests based on changes to mission requirements and the emergence of differing Conditions of Interest.[5] We now come to the information portion of our model.

The Query Planner sends the requests (*r*) to all of the Information Directories (ID) it is aware of (i.e. all those available to it on the GIG.) The Information Directories serve as interfaces to the Information Stores (IS). It is easier to think of the information Directories as the Yellow Pages of the GIG. They contain information about the particular sources of information and what they possess or can provide. The Query Planner sends the requests to the Information Directories who forward them to the Information Stores; which in our example are the Weather Information Store (IS_1 in our diagram) and the Shipping Information Store (IS_2 in our diagram). These ISs receive the request, check the information they have available and submit it back as the response to the respective query.

5 Conditions of Interest define critical parameters a user asserts could have a negative impact on the success of the mission. The user defines what values or changes in values should cause the VIRT system to alert the user. (RHR, 2005)

In this case the weather and shipping information requested flow back to the Carrier Strike Group ship(s) that requested it. Theory 1 doesn't seem so bad on the surface, which is good because it is arguably the way our current systems work (with some obvious variations). The problem, however, is that the cycle does not stop here. The sought-for, valued products (*v*) that are illustrated in Figure 1 do not precipitate magically. They are created from the information received (generally through a manual process or via an application running on the PE). There must be some filtering and massaging done to the volume of received data in order to change it into the desired product. This makes a Theory 1 process very inefficient.

Our personnel and organizations generally require small bits of specific information rather than all information about a particular subject. The problem is, under Theory 1, the only way to get the specific data is to get all of the data regarding a topic and then filter it somehow to make it useful. Take the example of our CSG transiting the Strait of Malacca. Currently, a ship receives meteorological assistance in the form of weather reports that do in fact take into account the intended track of the ship. As long as the ship transits along its submitted Position of Intended Movement (PIM) track then the system generates the appropriate messages containing the weather forecast to the ship.

The problem with this method is that the weather messages contain ALL the weather information, regardless of changes, every "n" hours (or minutes). This means on a periodic basis the ship receives a text-based weather message containing information that the ship's crew must decode, process, and format into a cohesive and common-sense visual representation of the data received. It is the ship's crew that ultimately filters all of the information and presents only what is germane to the chain of command. This consumes not only manpower, but wastes scarce resources in the form of bandwidth, processing power and time. Perhaps most important is the fact that significant events immediately affecting the ship can be overlooked or not be intuitively obvious, because they were lumped together with all of the other innocuous information. This can have devastating consequences.

A ship in the middle of a hostile area faces similar but heightened concerns. In tense situations that require split-second decision making, the Commanding Officer (CO) must have all of the *significant* information possible in order to make the correct decision. The CO does not need all of the information, just that information pertinent to the decision at hand. That is one of the issues we have now. Information from multiple different information sources is received. Until now, the primary focus has been on combining the information from those sources into a Common Operational Picture (COP) in an attempt to fuse the information and provide a better decision making foundation. In other words, let's take all of the information available from all of the sources we can connect the ship to. Then we process that information with an emphasis on fusion and making a better visual representation of it. And finally, we expect the user (i.e., operator) to filter out any information not desired after it gets there.

While this is a good start, it simply bandages a symptom rather than fixes the problem. That problem is that the system (in the form of personnel, programs, or equipment) must still filter all information *after* it is received, thus placing additional load on the people in the system. The CO must determine what information matters, and then act accordingly. In the case of an inbound missile, that may be a matter of 2-3 minutes. Seconds spent separating valuable information from information chaff will mean the difference between life and death. While the above may seem a bit dramatic, I can assure you that to the men and women who put their lives on the line, anything that can be done to ensure the effectiveness of that decision-making process will be most appreciated. Again, not all information is useful information. Theory 2 takes this observation into account.

2. Theory 2 – Smart-Push

What makes Theory 2 fundamentally different from Theory 1 is the way information is processed and allowed to flow. Take a close look at Figure 5 below. It is very similar to Figure 3 (Theory 1 process model), but there are a couple of significant differences. In Theory 2, the information flow itself is essentially the same as Theory 1. Queries are initiated from the individual

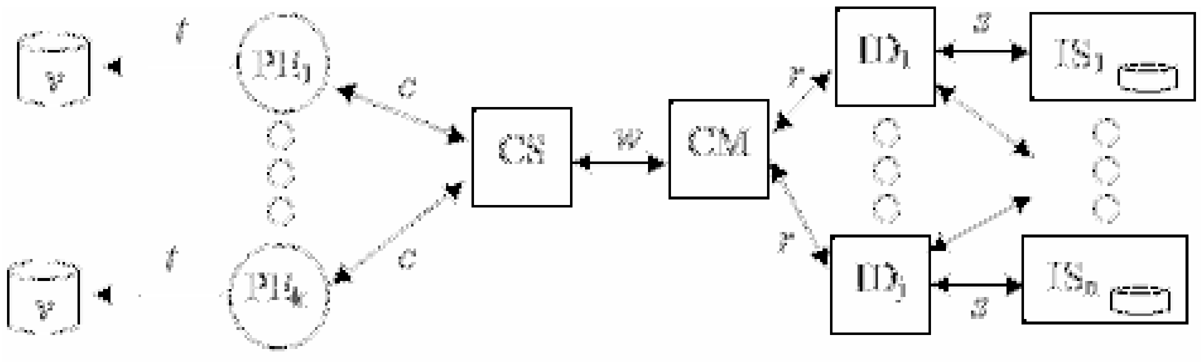

Figure 5. A value chain of processing entities PE_i producing products *v* as a result of specifying and monitoring Conditions of Interest (COIs) and then reacting adaptively to alerts. (From: Hayes-Roth, 2006)

units, processed against all information sources, and then the available information is provided back to the requesting site. The requesting site still takes this information and extracts some value (*v*) from it and uses it as needed. However, *all* of the information that was returned, in theory, will be useful (meaning relevant) and significant (meaning important). If it is not relevant and significant, the information will not be sent or received. Here are the highlights.

There are two boxes that are different between the two diagrams (Figure 3 – Theory 1 and Figure 5 – Theory 2) and these boxes completely change the concept of information-sharing. In the Theory 1 example of our fictional CSG transiting the Strait of Malacca, the query request was submitted, the results were received, and the entire cycle had to repeat itself, or at acceptable intervals in order to ensure the requesting ship or aircraft had the most current

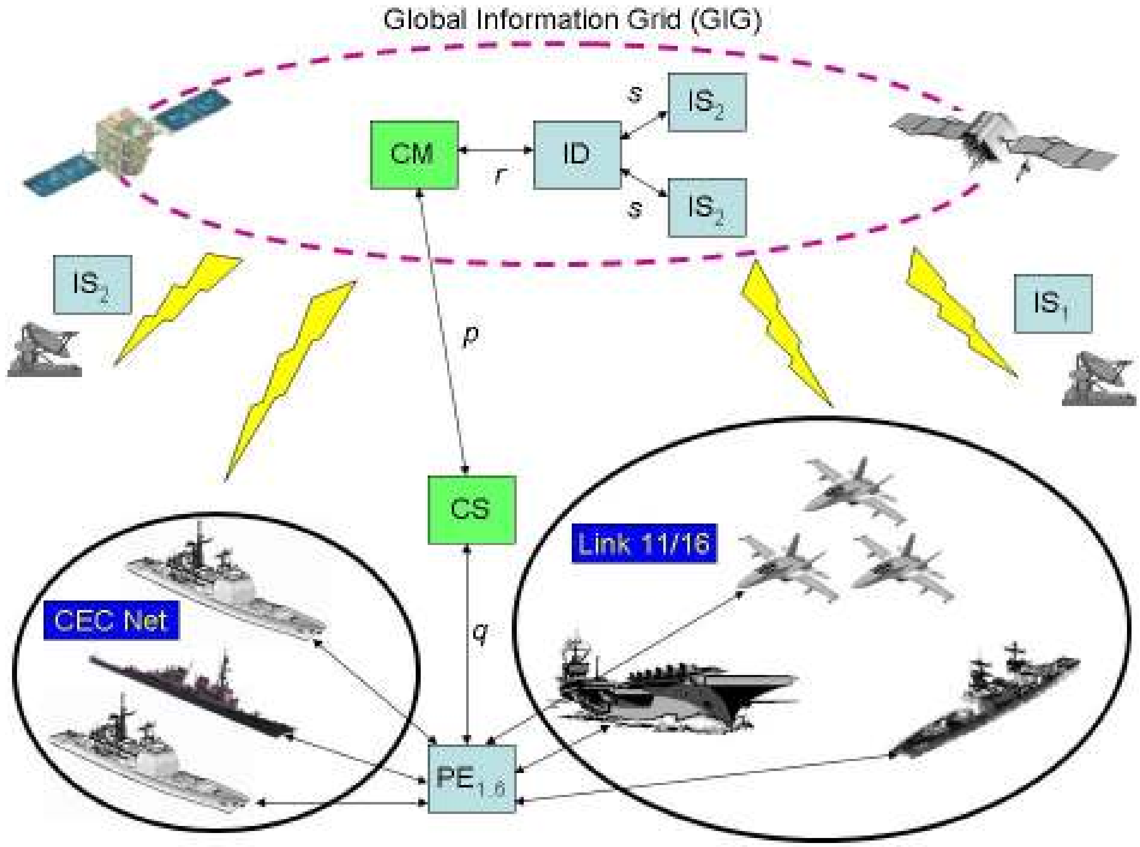

Figure 6. Real World Theory 2 Implementation

information. Replacing the Query Specifier (QS) and the Query Planner (QP) are the Condition Specifier (CS) and the Condition Monitor (CM). This is the essence of a VIRT-enabled architecture. For our fictional Strait transit the request for information would begin differently. This VIRT-enabled, Theory 2, real world implementation is shown in Figure 6.

The PE's still submit their request for information (weather and shipping in our example). However, the Condition Specifier is very different from the Query Specifier in that the Condition Specifier tells the system not just what information it wants, but how that information impacts their mission readiness. So not only would the CSG PEs send the request for weather in this case, they would also send their sea limits[6], course and intended movement (PIM), and any other special considerations such as desired route (in case inclement weather such as

[6] Ships have safe sea limits assigned as a function of design. These include the maximum wave heights and sea state a ship can take on the bow, beam, and quarter.

hurricane or typhoon requires a redirecting and change of course). Additionally, instead of just requesting shipping traffic information they can indicate a request for any high interest merchant traffic that they will encounter, or more importantly they might specify an amount or density of traffic that, if exceeded, would make their intended route undesirable. It is then up to the Condition Monitor (CM) to check continually for those desired conditions.

Now the Condition Monitor (CM) takes over and requests information from the IDs and ISs just as before. Only this time they do not immediately reply with generic information about the weather and shipping. Instead, the system is smart enough to realize, based on the conditions specified, what information will impact the ships. It has a concept of what information is valuable to those that require it. So “the system” sits and waits until it receives a piece, or pieces, of information that meet the requested criteria.

Suppose for example, the METOC IS has just downloaded new imagery from the weather satellite that indicates the presence of a typhoon outside the entrance to the Strait of Malacca. It compares both the location and predicted path of the typhoon with that of the CSG and determines that the CSG’s PIM takes it inside a point where it can be severely impacted by the storm. It immediately returns this information back through the system to the PE. The product of this information, (*v*) is not just a generic regurgitation of common weather info. Instead, what is sent is specific information regarding the storm such as where, when and how it will potentially impact the CSG. In addition it may include a recommendation for an alternate route. As discussed above, this alternate route should take into account the predetermined condition that indicated what routes are objectionable to the CSG and would recommend an acceptable path.

This is the beauty of VIRT. It assumes that different information has different value depending on the user’s dynamically evolving context. That value is unique to the particular user of the data and is customizable through the Condition Specifier. So not only does the system only allow the high value bits to

flow, it effectively prioritizes the flow of bits based on their value. Let me be clear here. What that means is that if a particular user requests multiple pieces of conditional information and only has a pipe of a certain bandwidth capacity, the system can be smart enough to dynamically determine the order of the bits going to the user and allow them to flow in a prioritized order. When resources are constrained, highest-value bits flow first.

III. CEC AND VIRT

A. COOPERATIVE ENAGAGEMENT CAPABILITY (CEC)

1. What is CEC?

On May 17, 1987 an Iraqi MIRAGE F-1 shot two Exocet anti-ship cruise missiles into the USS STARK (FFG-31). The F-1 had mistaken the STARK for an Iranian oil tanker and fired upon it. The result was 37 dead and 21 injured on the frigate. The STARK did not detect the missiles or aircraft in sufficient time to take evasive action. The reaction throughout the military and all levels of government was expectedly terse. We needed to ensure something like this did not happen again. One of the primary reasons for the STARK incident is that cruise missiles have become increasing complex and difficult to defend against. The missiles that hit the STARK were capable of traveling near Mach 1 (~662 nautical miles (nm) per hour.). They had a range over 100 nm and were sea-skimming. (MBDA, 2006) Sea skimming missiles are extremely difficult to detect with traditional radar systems because they travel close to the surface of the water (some as low as 10 ft). This means that the range at which a radar system can detect the inbound missile is significantly reduced. Additionally, this assumes that the radar system can differentiate the missile from the numerous waves and sea clutter that it also detects. The result is very little (if any) reaction time to allow the crew to take action to avoid being hit by the missile. One answer was of course to build better radar systems. While this option was, and still is, explored and new radar systems are continually developed, upgraded, and fielded, it is very hard to beat the principles of physics. The Earth is curved and objects traveling very close to the surface of the Earth are therefore extremely hard to see using standard radar waves traveling in a horizontal direction. One answer that was developed was called the Cooperative Engagement Capability (CEC).

While CEC wasn't developed specifically because of the STARK incident, its development was accelerated due to the event. CEC was originally conceived in the mid 1980's by Johns Hopkins Applied Physics Laboratory. (Walsh, 2005)

It was eventually handed off to Raytheon systems and has been improved over the past 15 years. The result is the most advanced and effective air tracking network currently available, anywhere! CEC provides three major areas of functionality.

> First, it enables multiple ships, aircraft, and land based air defense systems to develop a consistent, precise, and reliable air-track picture. Second, it allows combat system threat-engagement decisions to be coordinated among battle group units in real time. Third, CEC will distribute fire-control-quality targeting information, when available, among units in the force so that one ship or aircraft might be able to engage threat aircraft and missiles even if it does not have targeting data on its radars locally. (Bush and Grant, 2003)

What that means is that CEC allows multiple ships to share a “ground truth” air picture of sufficient quality that a ship’s fire control system can develop a firing solution on data that its radar systems don’t even hold. In addition, *all* ships hold the same information on all tracks. It would be an enormous mistake to think of CEC as simply another data link system such as Link 11 or Link 16. CEC is generations more sophisticated and more powerful than both of those systems. Traditional links such as 11 and 16 do not transmit radar data. Instead, a ship’s own systems generate a track based on contacts that are held on local sensors such as organic radar systems (a good example would be the AEGIS SPY1 radar onboard certain destroyers and cruisers). All of the individual tracks are then sent out via the network. The type of data link determines the extent of integration of the picture that results from the confluence of tracks from the other ships. In most data links, it is up to the processors on each ship to receive the other tracks and correlate them to sensor data held locally. In other data links a single participating ship correlates the track data and then broadcasts it to the others in the link.

All of these other link-driven systems have a couple of things in common that make them inferior. Chief amongst these deficiencies is that the link systems are sending tracks vice true radar data. This introduces error and ambiguity. How does a system know which track is more accurate? What

happens when the system receives multiple tracks that it cannot correlate? And lastly what does the system output when it holds radar data but the link information it is receiving does not match what it sees on its own sensors? The answer is that it either combines multiple tracks into a single track, or it inputs what is known as a dual track.[7] If the system mistakenly combines – or correlates – two tracks into one that are in reality separate contacts, the result is a lost contact and the ship can be vulnerable. If, on the other hand, the system creates a dual track there are now two tracks shown in the system but only one contact. Both of these conditions make the system extremely operator-intensive, because it is the operator that ultimately looks at the tracks and determines which ones are correct and which ones are not and then tries to correct the picture. Link 11 creates 1.5 tracks for every true track reported in the link. Link 16 is better by providing only 1.35 tracks for every true track. (Rivers, 2001) However, we can quickly see how multiple tracks can introduce problems into the system.

This is the primary reason a ship has, until now, been unable to fire on a track it does not hold radar information on. This is the major difference between the standard traditional link system and CEC. CEC does not submit tracks to the network of ships, aircraft and shore stations. Rather, CEC employs "multisensor measurement fusion as opposed to single sensor tracks to allow battleforce-centric, rather than platform-centric engagement." (Rivers, 2001) In other words, CEC sends raw radar data through the link. This radar measurement data is what allows the CEP (Cooperative Engagement Processor) to correlate the data and display a single composite track. The notion of a composite track is what makes CEC powerful.

CEC is a closed network system where all participants use the same algorithm and share the exact same air picture. The more participants, the better the picture gets because multiple views allow for better refinement of a track.

[7] A dual track is the result of the link inserting a separate track when it is unable to correlate the contact with existing tracks. The result is two link tracks for a single contact. This is known as a dual track. (Erwin, 2001)

This is because there are many things that affect the radar picture a ship receives. Weather, sea state, and altitude all affect the resulting radar picture received by a ship's sensors. Not all tracks are held at the same time, and for the same amount of time, by all platforms. By combining the raw radar data of multiple ships into a single network and creating a composite track out of the pieces that each participant holds, we get a much more refined and complete track as shown in Figure 7.

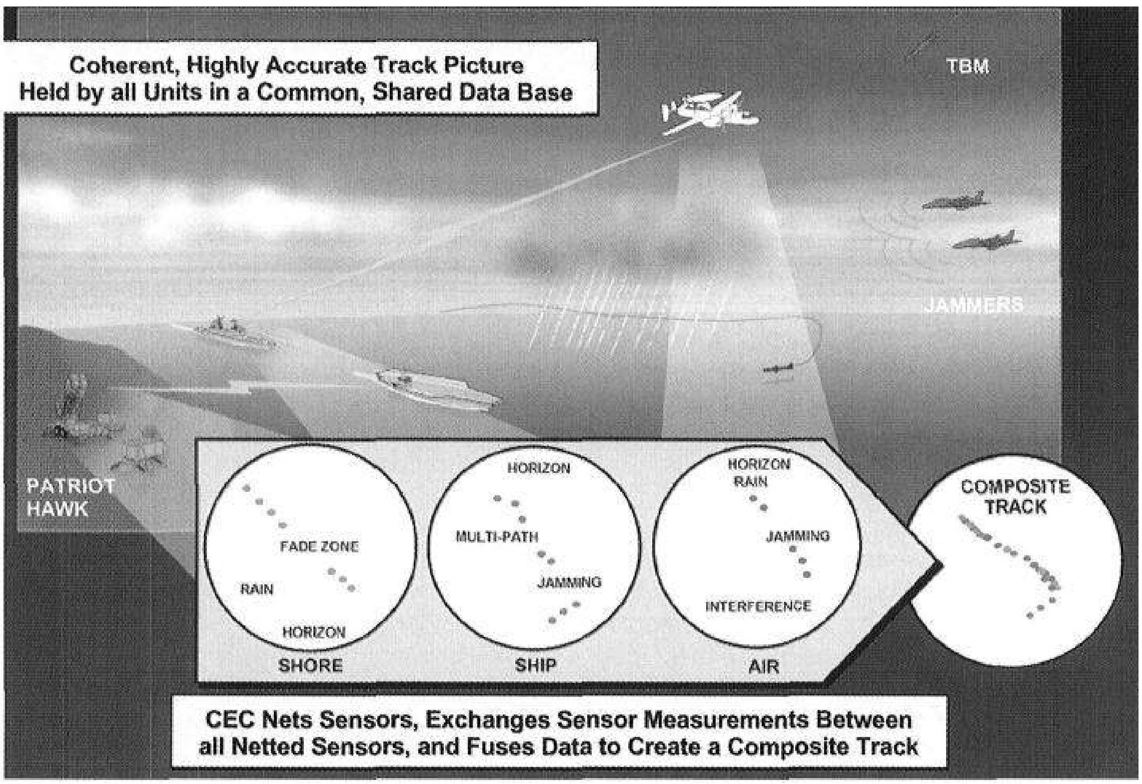

Figure 7. CEC Composite Track Fusion (From Busch and Grant, 2003)

Take for example a carrier strike group operating in enemy waters. One of the destroyers detects a sea-skimming cruise missile inbound to the group. However because of the proximity of the missile to the ocean, the radar system only gets a few radar "hits."[8] These few hits do not provide enough information to allow the fire control system to maintain track or create a fire-control solution. The aircraft carrier and one of the cruisers in the group also detect the missile a

[8] A radar hit as discussed here refers to an instance of time where the radar holds contact on a particular track.

few seconds later and both have the same issue[9]. The missile is simply traveling too fast and too low to maintain track. At this point, the likelihood of successful engagement of this missile by the strike group is not high. Now assume the CSG is CEC capable. The bits and pieces that all three ships hold will be combined to create a composite track that all three can use to fire on the incoming target, thus increasing the likelihood of a successful shootdown.

The next major deficiency of traditional link systems is the requirement for a ship or unit to maintain the link. CEC requires no such link or net control unit. This provides the primary benefit of survivability. A ship can join or leave the CEC network without compromising link integrity. Of course the more ships, aircraft, or other units that are participants in the CEC network, the better the picture will become, because CEC provides a composite track based on the data held by all units. Withdrawing from the link, while potentially reducing the overall accuracy of the picture, does not bring down the network.

2. CEC and VIRT

VIRT is the natural progression and the enabler to take CEC into the future. As originally designed CEC was intended as a Navy only system. As with most systems developed within DoD, it used proprietary hardware and CMS-2 software which made it extremely difficult to integrate or adapt with other services and systems. In today's world systems simply cannot survive without being able to adapt to new environments, missions, and operational concepts. CEC is no exception. If CEC is going to continue as a viable system, it must be made joint. The Program Executive Office for Integrated Warfare Systems (the Navy organization responsible for the CEC program) is fully aware of this. In fact, many changes have been made over the last several years in an attempt to move to an open architecture. Proprietary hardware such as processors has been replaced by commercial off-the-shelf available items. In addition, instead of CMS-2, new software is developed using C++ - a much more universally accepted standard. (Walsh, 2005) The size of a CEC install has always been a

[9] There is CEC implementation for an aircraft. This is primarily put on Navy E-2C aircraft. These installations maintain the same functionality as the surface installation.

deterrent to adoption by other services. The fact is that although our joint brethren realized the potential CEC had to offer, the equipment made adoption of the system difficult. Raytheon is trying to change that with the development of significantly smaller systems. A complete install of a standard CEC system on a ship weighs 3,614 pounds and an aircraft install is 688 pounds. In addition, both systems have a large footprint. The new systems being tested are down to 1770 pounds for a surface ship and 520 pounds for an aircraft. These reductions have been accomplished without any decrease in performance. (Fein, 2005)

So over the last 15 years CEC has gone from a proprietary system to one that uses an open architecture. Additionally the system has shed weight and become streamlined in attempt to make it viable for a broader class of users. It only makes sense then that the next phase of evolution for CEC should be a VIRT-enabled focus on sharing information with other agencies, communities, service, and countries.

a. Benefits to CEC as Consumer

VIRT has the potential to provide a benefit to CEC both as a consumer and as a supplier. In order to talk about the benefits we must first look at the GIG. The reason is that the GIG, as previously mentioned, will be the backbone of future information-sharing systems. Therefore, any discussion about enabling information-sharing will be incomplete without first looking at how the GIG will impact both information-sharing and the development of information-sharing systems. As discussed earlier, the GIG is DoD’s answer to providing this functionality for the future. As envisioned (see Figure 8 below) the DoD response relies on metadata and markup as the answer to enabling interoperability and information-sharing. It is this heavy reliance on XML markup that, in my opinion, represents a significant pitfall.

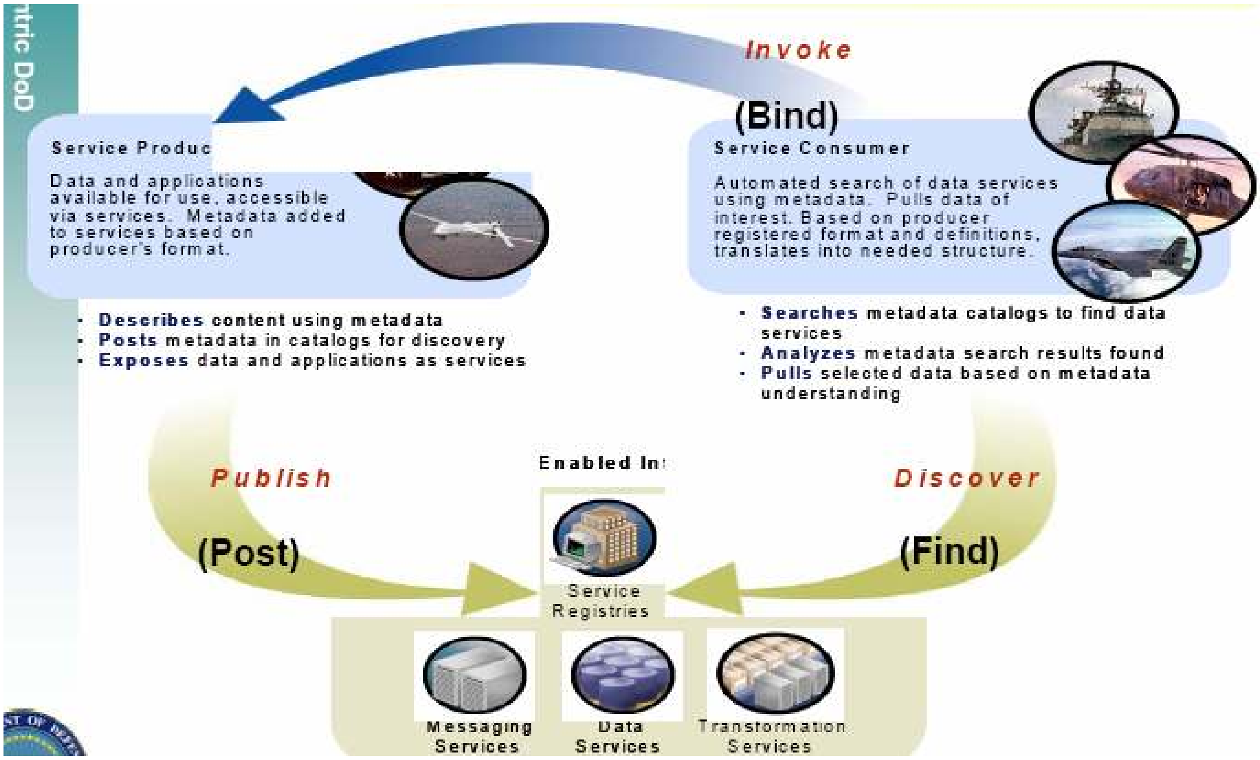

Figure 8. Vision of the GIG (From Robinson, 2004)

The use of XML tags and meta-data was mandated by Deputy Secretary of Defense Paul Wolfowitz in 2004. (DoD Directive 4630.5, 2004) Although XML mark-ups alone are not the complete answer for enabling interoperability and increasing information-sharing, this approach marks a *major* departure from our traditional short-sighted way of dealing with emerging information technology. This approach is a good start. However, in order to make the best use of the more than 10 Billion dollars the development of the GIG is likely to cost, we must go farther, more quickly. In addition to XML markup, we need to provide a VIRT-enabled infrastructure and also address the semantic interpretation of the information that will be shared.[10]. Metadata provides a way of cataloging information that will make it easier to find, so multiple users in different organizations should be able to access and employ it. However, the *meaning* of

Semantics refers to the interpretation or meaning of specific words or language. Dicitionary.com, 2004.

the data is just as important. The term *track*[11], for example, means different things to different users. A common semantic framework is required to ensure that the information provided is what was expected by the user who made the original request. A good example of this is *pizza*. If you take a trip Naples, Italy and ask for a *pizza*, you will get a *pizza*. Chances are though, that it will not be the pizza you are accustomed to or were expecting to receive. The reason is that the semantics of *pizza* are different based on region of the globe.

How can the implementation of a VIRT architecture help CEC as a consumer? The primary way VIRT can help CEC as a consumer is by reducing bandwidth demand from other shipboard applications that are likely to compete with it. As originally designed, CEC operates in the C Band line of sight. Further enhancements can provide satellite functionality for the CEC network traffic. However, the likely means of connectivity for CEC in the future will be TCP/IP via SATCOM. One good reason for this is that nearly every ship in the Navy is now TCP/IP-enabled through Super High Frequency (SHF), Extremely High Frequency (EHF), or INMARSAT.[12] It makes sense that future implementations of CEC will take advantage of this pathway [TCP/IP] to increase network participation. As discussed previously, the more nodes in a CEC network, the better the information available to all participants. However, if CEC is using bandwidth it will have to compete with the other shipboard applications also demanding bandwidth. It is unlikely that the information shared within the CEC network will be VIRT-enabled.[13] We can, however, reduce the bandwidth required for the other systems through VIRT.

A ship runs many non-CEC, bandwidth-demanding, applications. Medical, administrative, professional, morale, intelligence and environmental

[11] A semantic *track* model is the focus of Professor Rick Hayes-Roth in his 2005 paper titled "Towards a rich semantic model of track"

[12] IT-21 is the Navy's program that brings a suite of computers and applications installed on surface ships designed to enable enhance tactical and administrative capability. One component of the installation is INMARSAT which brings TCP/IP connectivity. All ships will have full IT-21 capability by FY 07. (O'Rourke, 2005 http://www.history.navy.mil/library/online/navy_network.htm)

[13] CEC is a closed network system that requires every participant to pass all its information over the network. Filtering via VIRT would adversely affect the way the system correlates tracks based on bits of information from each participant.

systems all require bandwidth. Decreasing bandwidth required for those applications by placing a VIRT service on the GIG can free up bandwidth for the CEC ship.

b. Benefits to CEC as a Supplier

When CEC is supplying information to other systems, VIRT has the most potential for benefit. CEC was designed to provide a near perfect air picture to every participating member in its own network. It has successfully accomplished that goal. It's time to take it to the next level. As I said in the Chapter 1 – information must be shared. The more people, organizations, and services we can share information with, the more powerful that information becomes. There was no single "smoking gun" that foretold the events of 9/11. There were, however, as detailed in the 9/11 commission report, multiple pieces of information held by various agencies and departments. Separately these pieces of information were of little value. Together, however, they were potentially a significant predictor of events to come. Timely delivery of that information to interested analysts would have been critical, too. The information CEC can provide can be one of those pieces of information. We need to make that information available to the other services, governmental entities, and coalition partners who would value it.

VIRT addresses the question of how CEC provides that data to external users. Remember that a Theory 2 – VIRT-enabled process – requires that information be provided in relation to expected mission impact. In the case of CEC, its data can be seen as being used by three distinct classes of users; the agency or organization (such as the Department of Homeland Security's Air Marine Operations Center (AMOC) or another service such as the Air Force), non-CEC equipped strike group surface combatants, aircraft and coalition members, and the dismounted infantryman/SEAL. Each user has a set of specific mission information requirements and a finite amount of resources (i.e., bandwidth). The graph in Figure 9 helps illustrate the classes of potential consumers of CEC information. How does CEC share its information with all three classes mentioned above?

Consumer	Bandwidth Available		
DHS / AMOC	Shore-Based HIGH		
Carrier Strike Group / Coalition		Sea-Based MEDIUM	
Infantryman / SEAL			Mobile LOW

Figure 9. CEC Consumer Classes vs. Available Bandwidth

For an organization such as the Department of Homeland Security or other shore-based agencies, communications bandwidth is not the gating factor on overall performance. T3 and faster lines abound and if bandwidth is lacking, more can be accessed with minimal pain or expense. For these consumers CEC data can be sent and processed in its entirety. Remember, however, that the CEC unit itself does not have unlimited bandwidth and thus a VIRT-enabled Smart-Pull architecture will benefit not just the requesting consumer but also the CEC provider as well The importance of filtering is further amplified when we move down the chain to the other non-CEC equipped ships and aircraft. This is particularly true of our coalition partners who often lack the money or space to increase their bandwidth capability. For these vessels some filtering must be done in order for them to make use of the information available from CEC and also work within their bandwidth capacities. Some may argue that the traditional data links help perform the function of sharing this information. That is incorrect. Remember that the information provided by data links is nowhere near the same quality with respect to accuracy or time latency that CEC provides. I do not assert that a non-CEC unit would get exactly the same quality (i.e. fire control) information as a participating unit in the CEC network. However,

the information provided by the CEC is still vastly superior in terms of accuracy than that of currently available data links. This class of users cannot simply increase the available bandwidth with the ease of the first class. This brings us to the final class of users, the infantryman or SEAL on the ground.[14] Users in this class have significantly constrained bandwidth. They often use portable or handheld devices and share bandwidth with other theater organizations and users. This reduces the amount of bandwidth available to them and they require the filtration that VIRT can provide.

Personnel bandwidth, however, is a concern for all of the classes of users including the CEC ship. The Navy in particular has developed a new strategy to ensure that we adapt the force for future operations. It is called Sea Power 21. One of the main pillars of Sea Power 21 is Sea Enterprise (as shown in Figure 10 below). The reason Sea Enterprise is important to our analysis of

Figure 10. Se Power 21 Pillars (From Clark, 2002)

---

[14] SEALs often have access to state of the art equipment which can potentially provide them with increased bandwidth. For the purposes of this analysis I make the assumption that the SEAL's bandwidth is extremely limited.

bandwidth is because as the force strategy is implemented Sea Enterprise will likely affect personnel strength. The three tenets of Sea Enterprise shown in Figure 11 require that we strive to achieve the "right force with the right readiness

Figure 11. Sea Enterprise Components (From Mullen, 2004)

at the right cost." (Mullen, 2004). The right force and the right cost pieces mean restructuring manning. Although this does not necessarily correlate to a reduction in force strength, it will mean reallocation of personnel and an emphasis on minimum manning to reduce costs. The end result will be increased demands placed on our personnel. VIRT will reduce the load by providing the operators only with relevant and significant information.

If we look at the future of CEC in relation to Theory 2, CEC will be either an independent Information Store (IS) or an input to a separate IS. It is also entirely possible it could be both. The end result though is the same. CEC must find some way of filtering the information requested from it rather maintaining a continuous flow or periodic dump of all contacts held. This will make the information valuable for external consumers and non-CEC network participants.

B. AIR MARINE OPERATIONS CENTER (AMOC)

1. Description and Relevance

The Air Marine Operations Center (AMOC) is located at the March Air Reserve Base in Riverside, CA. It is operated by the Division of Customs and

Border Protection that falls under the Department of Homeland Defense. Its primary mission is as follows:

> Provide direct support to Homeland Security in protecting the American people and our national borders through the detection and identification of transnational threats and coordination of law enforcement air and marine forces. (AMOC Training Manual, 2005)

In short, AMOC personnel provide real-time monitoring of U.S. borders and airspace for instances of illicit activity. These activities range from drug smuggling across the border, to illegal immigration, to terrorism. AMOC then coordinates with local law enforcement and other governmental agencies to effect arrests and seizures as required. What makes AMOC unique is the underlying information systems that enable their mission.

The information system used by AMOC consists of commercial-off-the-shelf (COTS) hardware with proprietary fusion software. AMOC receives input from numerous different information sources and has the capacity for 450 separate radar inputs. AMOC's system takes these various inputs from sources such as DoD (shore-based), Federal Aviation Administration (FAA) and Aerostat and fuses them into individual tracks which are presented to the operator. It has the capability to process over 24,000 fused tracks every 12 seconds making it very robust. (AMOC, 2005) This capability should sound familiar. It has much of the functionality (though completely different in design and operation) as the CEC system.

The AMOC proprietary information system provides the best example of information-sharing in the civilian or governmental sector. This system enables AMOC operators to get the best information possible. The ability to fuse these sources is why we care about AMOC and what makes their system so important to the future of information-sharing.

2. Improvements by Using VIRT

VIRT can benefit AMOC as a consumer or as a supplier. AMOC is a shore station and because of this they have access to a near limitless data pipe (bandwidth). The benefit VIRT provides to AMOC as a consumer is not a

reduction in communications bandwidth. However, AMOC is minimally manned. They have extremely limited personnel bandwidth. Although the information system fuses tracks, it is still up to the operator to consult a variety of other sources (law enforcement databases, FAA flight plan data, etc.) in order to make a determination regarding the nature or intent of the contact. VIRT can mitigate this minimal manning by providing the existing operators with just the information they require, when they require it.

As a supplier, because AMOC receives information from a multitude of sources, they also have much information to share. Political and procedural boundaries notwithstanding, AMOC information can be valuable to other agencies and even military services. However, not all AMOC information will be seen as valuable to these other agencies. Therefore, a VIRT architecture can filter the non-relevant information out and provide each recipient very concise pertinent data. Under Theory 2, AMOC becomes an IS and is now an information provider to those who need it.

A Single Integrated Air Picture (SIAP) is the desired utopian end state with respect to air defense. (NDM, 2002) In principle, it provides a very clear picture and at the same time helps to prevent fratricide caused by improper identification of air tracks. Both CEC-equipped and non-CEC-equipped ships can benefit greatly from fusion of information from AMOC. The reason, as stated earlier, is that AMOC receives inputs from a multitude of sources. Most of these sources are not available to military forces – regardless of whether they are CEC-capable or not. AMOC can also benefit from receipt of the CEC air picture. The accuracy of the CEC air picture can enhance the quality of the information fusion at AMOC. The U.S. Navy routinely operates off the coast of the U.S. It is entirely feasible that while operating, these vessels can seamlessly and effortlessly send information to the GIG where AMOC can receive it if it meets the conditions of interest AMOC has indicated. For example assume that AMOC had submitted a request to the Condition Monitor to be alerted of any aircraft inbound to the West coast of the United States, traveling over 400 knots (nautical miles per hour) and

not squawking any IFF (Identification Friend or Foe).[15] When CEC cruisers detect an aircraft meeting those conditions and submit that information – automatically – to the GIG, AMOC can then use that information to aid it in the determination of whether or not this contact merits further scrutiny. Again, AMOC can both provide and receive benefit from a VIRT implementation of the GIG where they act as both a supplier and consumer of timely, relevant information.

[15] IFF is a system that allows identification of aircraft based on a transponder signal the aircraft emits.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SIMULATION METHODOLOGY AND TOOLS

A. SIMULATION AND TOOLS

1. Simulation

With the emergence of information technology advancements, modeling and simulation has become a very attractive option for experimentation. High speed computers and inexpensive, expansive storage capabilities have made it easier than ever. So what exactly is a model or a simulation?

> A model is a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system. A simulation is the manipulation of a model in such a way that it operates on time or space to compress it, thus enabling one to perceive the interactions that would not otherwise be apparent because of their separation in time or space. (Systems-Thinking, 2004)

Modeling and simulation are powerful tools because they allow a user to test out a theory or design process with software rather than creating physical mockups that are often time consuming and very expensive. Auto manufacturing is one prime example of how modeling and simulation are used. When automakers design a car, they use a computer model of the car to see how the parts work together and perform. Imagine how expensive it would be to design and build a new car only to then find out that the engine was not powerful enough to move the vehicle. Or perhaps the transmission doesn't mate to the engine. Modeling and simulation gives us the ability to test these types of processes before committing concepts to hardware.

One of the most common forms of modeling is using *Discrete Event Simulation (DES)*. DES works on the premise that the state of a system changes at discrete points in time, defined to be the occurrence of an event. We can use a model to simulate specific events at fixed or random intervals. For instance, if we run a restaurant and we are trying to find out how many service staff we need, we can use a model to find out how many people can be served by a specific number of wait staff. By using the time feature of a DES the model can

have people arrive at predetermined intervals – perhaps 10 people every 5 minutes. The simulation clock advances to the time of the next scheduled event occurrence. Depending on the event occurrences, a DES can run faster or slower than real time. In the previous example, simulation execution time of a few minutes could provide restaurant data correlating to an entire day.

This brings us to *event graphs*. Event graphs are a graphical representation of a DES. Event graphs help visualize design of a DES by showing events as circles and event scheduling as arcs or edges connecting the circles. The benefit is that this simplifies development of the model. The problem with event graphs and other modeling techniques is that they can be complex and time consuming. Tools have been developed to facilitate model design and implementation.

2. SIMKIT and VISKIT

"SIMKIT is a collection of Java libraries that support implementing event graph models." (Buss, 2004). It provides a Java Application Program Interface (API) for implementing a DES. VISKIT is a graphical user interface (GUI) for describing a DES using the event graph notation. VISKIT auto-generates a Java implementation of the model using the SIMKIT API. VISKIT allows those with modest programming and simulation skill to design and implement an event graph model. A sample screen shot of the VISKIT user interface is shown in Figure 12 below.

B. METHODOLOGY AND MODEL DESIGN

1. Overview and Considerations

The primary purpose of the modeling and simulation conducted in this thesis is to measure bandwidth utilization. The intent was to demonstrate, through experimentation, the hypothesis that Theory 2 can provide a measurable bandwidth advantage over a Theory 1 implementation. The models were built to that end. Additionally, the model provides an initial implementation of logic to compute processing utilization of the user PE. Though not the focus of this thesis, I was curious as to whether or not the processing required for a Theory 2 system would be more or less than that of Theory 1 in this implementation.

Since Theory 2 responds only when required I hypothesized that the processing requirements placed on the system would go down as well. The PE and CS, for example have to process fewer inbound and outbound queries, potentially freeing up processing (human or machine) resources.

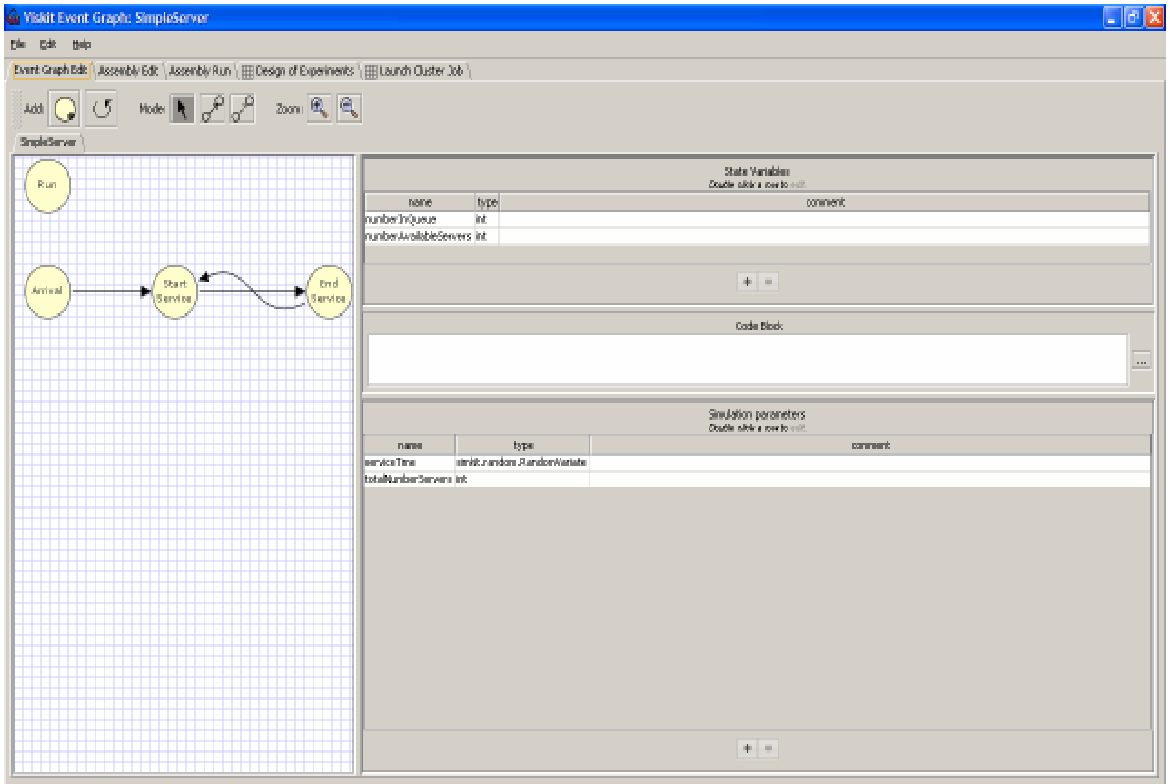

Figure 12. VISKIT Screenshot

2. Theory 1

The Theory 1 model is designed as a fairly simplistic deterministic model of the query/response mechanisms. The Theory 1 model relied on the Smart-Pull premise that a ship (or any other PE) initiates a query of fixed length at specified, recurring intervals. This approach is consistent with the Hayes-Roth example of the aircraft pilot in his paper on the Two Theories. (Hayes-Roth, 2006) The simulated distributed information "system" automatically generates a response to each query. This is because under Smart-Pull, the system provides the requested data back to the user every time the request is made.

The basic event graph model of the Theory 1 concept is shown in Figure 13. This is essentially a 4-stage-service model. The user PE periodically

initiates a query. The processing resource (human plus computer) is utilized some period of time to create the query and the query message is sent over the network. The network resource is utilized for a period of time to transmit the message. The information system receives and responds to the query (events titled “Network Receive Query” and “Network Send Response” in the figure), again utilizing the network resource to transmit the response message. Finally, the user PE processes the response utilizing the processing element resource.

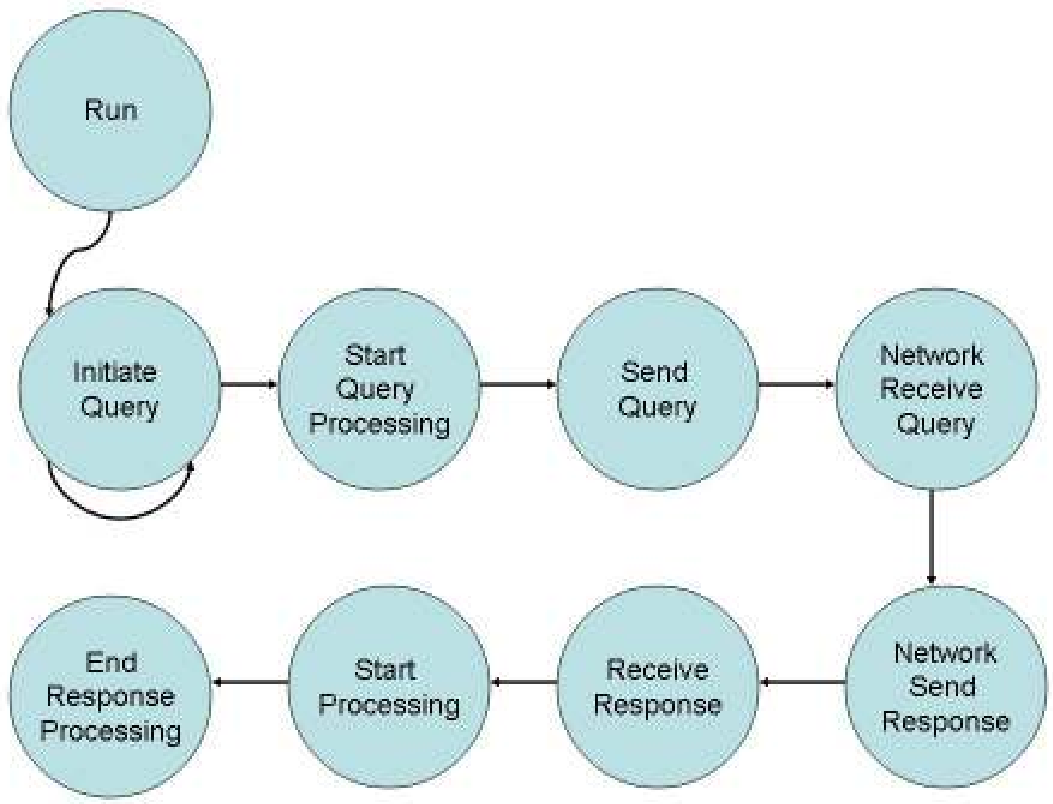

Figure 13. Theory 1 VISKIT Design Model

The caveat is that the user must continually request, or pull, the data from the system. This is why, as illustrated in Figure 13, that the start of one query schedules an event (the self-scheduling edge in the “Initiate Query” event) for a follow-on query. It was my intent to remain as true as possible to the scenario described by Hayes-Roth for Theory 1. The following parameters and justification were used:

- Query Time: Query Time is the average processing time to generate a query. This will be a function of the processing capability in the notional system that is generating the request. This processing time was of secondary interest in the model, and was provided just as an initial approach to modeling the user-side processing demands of the two theories. This value was set at a constant value of one second for my Theory 1 model.
- Query Length: Query Length is the size of the query message in bytes. For a Theory 1 instantiation of the model, the actual size of the message was not as important as the fact that the message size remained the same throughout the run. This is because in a Smart-Pull process design, the user will generally request updates to the same pieces of information on a continual basis. For example, in the case of an aircraft pilot, he will want to know weather information. He will continually request the same weather information as long as he is conducting the same mission. This is because Theory 1 has no way of determining what information actually has an impact on his mission. Therefore, he requests it continually at predetermined intervals to ensure he has the most up-to-date information. The value chosen for Query Length was 1KB.
- Query Interval: For the query interval a time of 10 minutes was used. This means that once every 600 seconds a query is generated by the system. Again, this repeated querying is necessary because of the design of the Smart-Pull process. The user only receives a response after the information request has been generated and then received and processed by the information system.
- Response Time: The response time is the time required for the system to generate a response once the query has been received.

The time to generate a response was not important to the study. For this reason a nominal response time of one second was chosen for this model. Future model enhancement can explore details in the information system architecture regarding the query processing components (Query Processor, Information Directory, Information Store), but this detail had no bearing on the current study.

- Response Length: A response length of 500KB was used to represent the message size that was sent by the system to the PE every time a query was received. As with the majority of the numbers for this Theory 1 model, the value was chosen based on the Hayes-Roth paper (Hayes-Roth, 2006,) and represents the portion of the full data set that is returned to the user in response to his query (and within which only “a tiny fraction of these [data] will justify a change in plan”).
- Transmission Speed: Transmission speed is a bandwidth-limiting parameter used to calculate the time the network resource is in use to transmit a message. A notional time of 1.25 MBps (10M bits per second) was chosen to represent the network transmission speed. (Recall from Figure 2, Chapter 1 that the projected bandwidth capacity for a command ship in FY07 is approximately 10Mbps.)
- Processing Time: Processing time is the time it takes for the system to process the query, modeled as seconds per byte received. The longer the query response, the longer it takes the system to process it. The value used in the calculations for the Theory 1 model was 10 microseconds per byte.

3. Theory 2

Theory 2 presented much more of a challenge. Although many of the parameters and events modeled were the same as in the Theory 1 model, the implementation approach was changed to better represent the dynamics of the Theory 2 Smart-Push concept. Figure 14 illustrates the Smart-Push event graph

model. In the Theory 2 model, the network schedules a response according to an exponential distribution vice the self-generating response design of Theory 1.

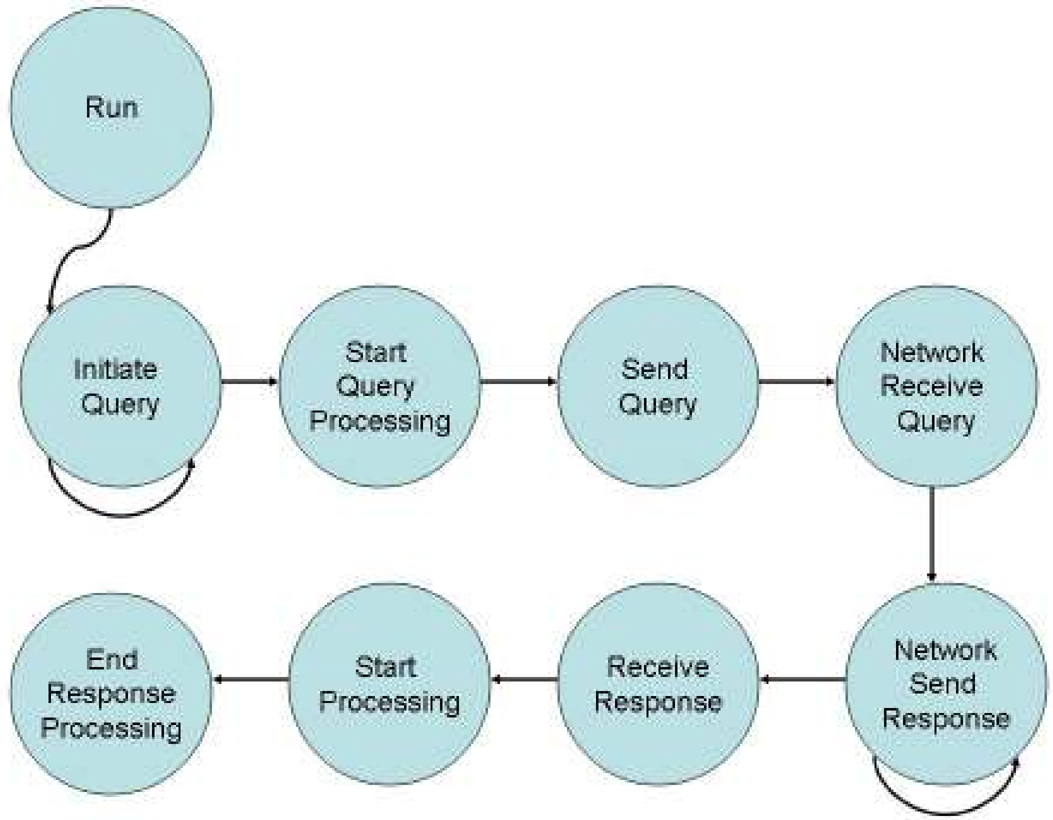

Figure 14.   Theory 2 VISKIT Design Model

There are three major differences that had to be addressed for the Theory 2 implementation:

a. Number of Responses

You'll recall that the Theory 1 model generated one and only one response to each query. For Theory 2, it is more reasonable for the information system to retain the query (actually, the user's conditions of interest) and to respond any time the battlespace situation is deemed to satisfy one or more of the conditions of interest. Therefore, the model has to initiate a response or set of responses to the query submitted by the PE. The number of responses varies, just as it might in the real world. The CM will respond to the PE when it has information that meets the desired Conditions of Interest (COIs). For this initial model implementation, there was no list or database of COIs nor was there a

representation of changing battlespace state that would cause a condition of interest to explicitly be met. So the question was how to simulate the number of times the system would respond to a particular query. The answer chosen was to use a Poisson distribution. A Poisson distribution "expresses the probability of a number of events occurring in a fixed time if these events occur with a known average rate, and are independent of the time since the last event." (Wikipedia, 2006) In other words, this approach allowed us to generate a random number of responses to a single query. The Poisson equation and generic graph (Figure 15) are provided below.

$$f(k;\lambda) = \frac{e^{-\lambda}\lambda^{k}}{k!},$$

where λ is the average number of occurrences per unit time.

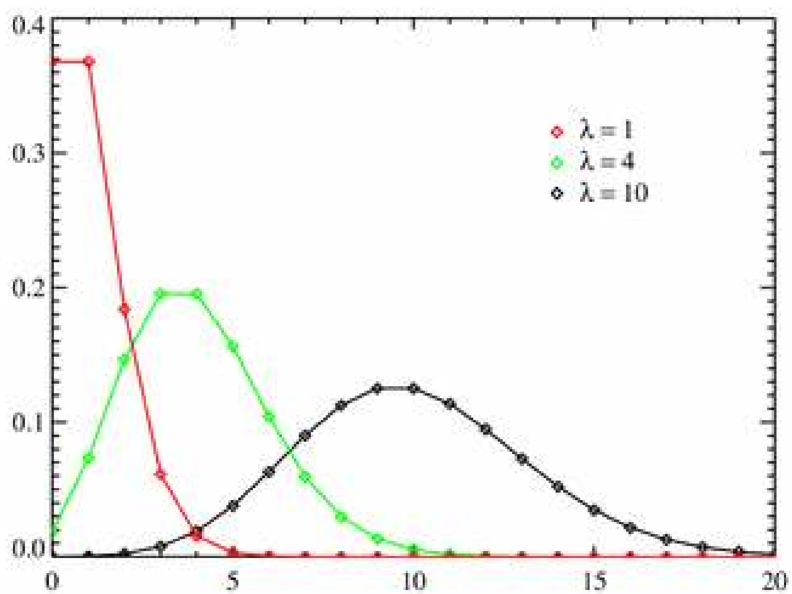

Figure 15. Poisson Distribution Graph (From: Wikipedia, 2006)

b. Time Between Responses and Time Between Queries

Theory 1 queries and responses were in a one-to-one correspondence. This meant that the system responded as soon as possible to each query. For Theory 2, not only would the number of responses to a query likely vary, but the frequency, or time between responses, would vary as well. A

Theory 2 process does not receive an immediate response to a query. This is because the system only responds when there is *relevant* and *significant* information for the user regarding mission accomplishment. In fact it may be some time (or not at all) before the querying PE receives a response. If no event occurs that satisfies the conditions of interest, no data need flow. Additionally, a PE only submits a query when it needs to update its conditions of interest. These queries are likely to be much more infrequent than the continual, periodic pull of Theory 1. The method chosen for the Theory 2 model to simulate the time between queries and the time between responses was an exponential distribution. This allowed us to randomly generate the times between queries (updates to conditions of interest) and responses (occurrence of critical events.) given an expected mean for each. A sample exponential distribution graph is shown in Figure 16.

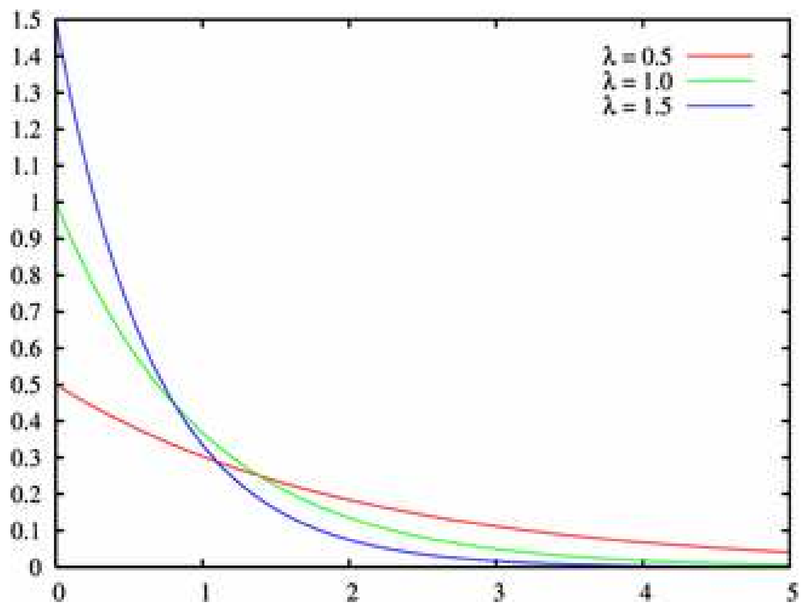

Figure 16. Exponential Distribution Graph (From Wikipedia, 2006)

c. Query and Response Lengths

Lastly, message lengths are not fixed. In Theory 1 we assumed the system responds back with a fixed message length. This is because notionally the user, under Theory 1, initiates the query for the same types of information repeatedly. The system responds accordingly with the same (perhaps updated)

amount of information. In Theory 2, however, varying message lengths is a more reasonable assumption. The system in Theory 2 is VIRT-enabled. Since only the pertinent information is passed to the user, the message length is likely to be shorter than that of its Theory 1 counterpart. Additionally, the response generated by the system will also be of varying length since the message returned will be focused, and based on the COIs of the PE that were satisfied by some battlespace situation or event. A Gamma distribution (Figure 17) was chosen to generate the message lengths required. One of the main reasons for choosing the Gamma distribution is that it does not allow any negative results since there could not possibly be a message of negative length. Additionally, the Gamma distribution can provide a relatively wide spread (high variance) with respect to values generated. This is important because message lengths for Theory 2 are based on the value of the message. Since there was no concrete way of simulating value for this model, the simulation assumed that messages of varying lengths will potentially be sent to the PE. Other distributions may be reasonable, especially when selection of an appropriate distribution can be driven by empirical data as VIRT research continues.

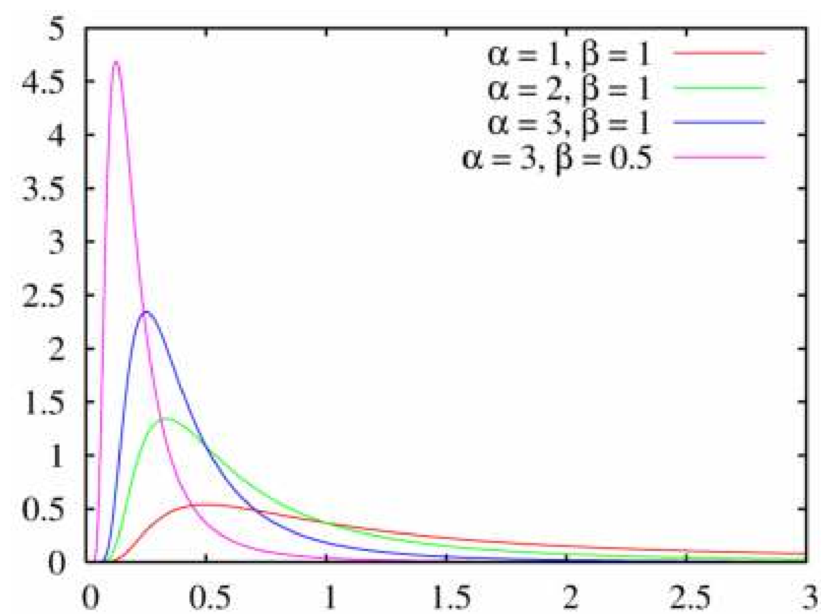

Figure 17. Gamma Distribution (From Wikipedia, 2006)

Based on the above distributions and assumptions, the Theory 2 parameters are as follows:

- Query Time: As in the Theory 1 model, the average processing time to generate a query was set to one second, and was not of significant concern to the study.
- Query Length: The Gamma function was used to determine query length. This produced a wide distribution around the mean. The mean in this case was 1000KB. This means messages of random length were generated that conformed to the Gamma distribution and gave an average of 1000KB. It is easier if we think of this in context of a ship transiting from the East coast of the United States to the Arabian Gulf on a deployment. When the ship leaves homeport they would submit a set of COIs. This will be a small message requesting such information as weather impacting the route and traffic that will intersect the PIM track. This initial message could be larger or smaller depending on the number of COI's the ship needs to convey. As the ship enters the Mediterranean Sea through the Strait of Gibraltar, it is concerned with other things. Its mission is now to get to the Suez Canal and continue to the Gulf, so it submits another request or change to its COIs. These now include weather along the new track – from Gibraltar to the Suez, anticipated port visit information, merchant and high interest information along the track, and any suspected terrorist information in the vicinity of the Suez. This message will be larger than the original set of COIs in the initial query. This process continues until the ship is safely in the Arabian Gulf, at which time they have a new mission and submit a new set of COI's.
- Query Interval: As discussed previously, the query interval uses an exponential distribution to generate a query at random intervals. In this case the mean chosen was 6000 seconds, vice 600 seconds

between queries for Theory 1. In the case of the Theory 2 model the *average* time between queries is assumed to be far less than that of Theory 1 because queries will only happen in response to a change in conditions of interest as opposed to a specified time interval. Because of the use of the exponential distribution in the generation of the query interval, this model could easily be modified to simulate multiple independent PEs making requests. By decreasing the query interval we effectively increase the number of queries because they happen more frequently. With respect to network bandwidth utilization, the result would emulate multiple independent PEs making queries at random times but with the same average time between queries (same basic frequency of need to change conditions of interest based on the mission and battlespace situation). Explicit modeling of different classes of PEs having different query characteristics is an area for future model enhancement.

- Response Time: The Hayes-Roth paper does not speculate on frequency of system response to a user's conditions, only that responses happen when the battle-space situation causes the conditions of interest to be met, requiring transmission of critical, valued information to the user. To model the randomness of the time between responses, I used an exponential distribution to provide a random response time that simulated the generation of information meeting a set of notional conditions of interest. The value chosen for the Theory 2 model was 100 seconds between responses to a single query.
- Number of Responses: When a PE generates a query and submits it to the CM, the CM waits until it has information that meets those COIs and then responds to the PE with the information and continues to monitor for any additional matches. It is probable that

the CM will send multiple responses to each set of conditions (query.) There had to be some way, therefore, to model the COI sending multiple responses back to the CE. A Poisson distribution was implemented with a mean of 25 to represent the number of responses the simulated CM would provide back to the initiating PE before assuming that the battlespace situation would no longer create events satisfying those conditions.

- Response Length: Response length will be less on average than for a Theory 1 query since the response will not have extraneous, non-relevant data. The Gamma distribution was chosen with an average response length of 1KB (compared to the constant 500KB response length in the Theory 1 model). Used in conjunction with the number of responses, these values provide an average of 25 responses with an average length of 1 KB per response provided to the PE.
- Transmission Speed: Just as in the first model, a notional time of 1.25 MBps was used to represent the network transmission speed.
- Processing Time: A processing time of 10 microseconds per byte was also used in the Theory 2 model, although this was not the area of primary concern in the study.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SIMULATION RESULTS

A. THEORY 1 MODEL

The two characteristics that were measured were processor and bandwidth utilization; although the latter is the output of primary interest for this study. For the Theory 1 model, only one run was necessary since this was implemented as a deterministic model. The results for Theory 1 were as follows:

- Bandwidth Utilization = 0.00066800
- PE Utilization = 0.01000000

B. THEORY 2 MODEL

Theory 2 measured the same characteristics as Theory 1. However, not all input values were constant but were modeled as random-variates based on particular distributions as described in Chapter IV. In order to obtain meaningful data, multiple runs had to be conducted. Statistics were gathered from a compilation of those runs. Sample means and 95% confidence intervals were computed from the outcomes of 101 runs of the model, simulating a period of length 60,000 seconds (16 hours and 40 minutes of simulated time).[16]

- Sample Mean, Bandwidth Utilization = 0.00000348
- 95% confidence interval, Bandwidth Utilization = [0.00000346, 0.00000350]
- Sample Mean, PE Utilization = 0.00020926
- 95% confidence interval, PE Utilization = [0.00020808, 0.00021044]

C. SUMMARY OF RESULTS

The first thing that should be apparent is that there is a large difference between the bandwidth used in the two models. As hypothesized, Model 2

[16] A confidence interval is used in determining the range of values an independent variable might take. In our example, the bandwidth utilization and processing utilization vary in each run. A confidence interval was developed that indicates the percentage of time the average output values will fall between the given values.

shows a decrease in bandwidth utilization. Under the assumptions and settings of the two models, the Theory 1 model used nearly 200 times the amount of bandwidth as Theory 2.

$$\frac{\text{Theory 1}}{\text{Theory 2}} = \frac{.00066800}{.00000348} = 191.95$$

Although of secondary interest in the study (since the processing side was represented in a very simplistic manner), it was also surprising how much the processing utilization went down for Theory 2. Processing utilization for Theory 1 was almost 50 times that of Theory 2.

$$\frac{\text{Theory 1}}{\text{Theory 2}} = \frac{.01000000}{.00020926} = 47.79$$

It must be emphasized, however, that these results only reflect assumptions and analysis presented in the Hayes-Roth paper and are not conclusive proof that VIRT can provide this much benefit. Further research and development work must be conducted to take the modeling and simulation to the next level.

Much refinement is possible in the simulation model design and setting of the input parameters. As further research is conducted, and as early implementations of the VIRT concept become available, empirical data can be collected to refine the simulation input data and modeling approach.

VI. COALITION INFORMATION-SHARING

A. THE STATUS QUO

We have discussed why VIRT is an enabler for the future of information-sharing. Additionally, I have shown that VIRT provides value in the form of reduced bandwidth over a non-VIRT implementation. However, in order to understand how VIRT can benefit coalition operations we must first look at the current state of allied information-sharing.

Since the revolutionary war the United States has relied on its allies and coalition partners to assist it through combat. It can be argued that some of these partners played more of a role than others, but they played a role nonetheless, and we have come to rely heavily upon allied participation and support. In addition, the number of coalition members we operate with has grown immensely over the last 200 or so years. In the revolutionary war we relied on help from the French. In World War I our allies included most of Europe. In World War II we operated with both the Europeans as well as many countries in East Asia. And finally in both Operation Desert Storm and Operation Iraqi Freedom the number of coalition partners reached as high as 32. In an era where we rely so heavily on multi-national operations, we cannot ignore the need to effectively and efficiently share information amongst them.

We do currently have systems with which to communicate with our allies. In the past, most of the information we shared was via voice network. These include VHF (very high frequency; bridge-to-bridge), HF (high frequency), and UHF (ultra high frequency) radios. Voice alone, is insufficient for today's complex coordinated operations so we allow many of our coalition members into our shipboard data links. These, however, do not allow the transmission of the various information we require such as images and e-mail. This was the primary reason we developed the Combined Enterprise Regional Information Exchange System (CENTRIXS). CENTRIXS is a vast improvement over previous information-sharing systems in that it allows the United States and coalition

members to communicate via e-mail using the Secure Internet Protocol Routing Network (SIPRNET). Additionally, it allows the transfer of other data formats. CENTRIXS, however, has one major drawback that prevents it from being very useful. It is as follows:

> Today, each CENTRIXS network is built to the same architectural standard but is not interconnected to prevent inadvertent release of data to nations who are not part of specific information sharing arrangements. Until sufficient accredited guarding technology exists, nations participating in multiple operations must maintain separate network terminals to ensure information integrity and confidentiality. (Boardman & Shuey, 2004)

This means that a separate terminal and network connection is needed for communication with each coalition member we wish to share information with. This is costly in space, equipment, and personnel. Also, as the number of coalition partners grows, the more limited this option becomes. Although this system is far from ideal, it represents the best we currently have.

B. THE FUTURE

As I have discussed throughout this thesis, I believe any discussion of the future of information-sharing must include VIRT. For reasons discussed above, the current systems for coalition information exchange are insufficient. This is important to the future of CEC because CEC units operate in conjunction with allied and coalition members. CEC ships and aircraft will provide information to our allies and will also consume information provided from them. The value of the information both supplied to and consumed from coalition members improves if the system is VIRT-enabled. The DoD understands there is a shortfall in the area of coalition information-sharing as well and has since commissioned an advanced concept technology demonstration (ACTD) to address this. COSMOS, or the Coalition Secure Management Operating System shows a great deal of promise. The reason is that it takes into account many of the fundamentals that I believe are essential to information-sharing and interoperability. The first priority is to use a common data model. This is similar to the semantic discussion we had earlier in the thesis. Step one has to be to make sure that everyone has a

common idea of what the data means.[17] COSMOS is addressing this. Secondly, COSMOS is using the GIG as a backbone for coalition sharing of information. This is a major improvement over current systems such as CENTRIXS and will be a significant enabler of a coalition information-sharing system. Coalition members do not have an information backbone such as the GIG and without access to ours information sharing would be inhibited. And lastly, COSMOS is trying to use technology (such as smart agents and enterprise services) to ensure that coalition members do not have to purchase large, expensive, single purpose equipment to be able to operate with U.S. forces.

Although COSMOS is a significant paradigm shift[18], it is not the ideal solution we should be heading for. With all of its potential, COSMOS is lacking one major concept that should be evident to the reader at this point – VIRT. The "shoot for the stars" solution must include a way to share information *smartly*. VIRT provides this smart capability a manner consistent with the service oriented architecture that is envisioned for the GIG. The best implementation of VIRT, and one COSMOS should take advantage of, is as a service that will run on the GIG and enable Smart-Push information-sharing to all who have access to the GIG.

C. HOW DO WE GET THERE?

So how do we get to this VIRT-enabled, information-sharing future? There are two near-term opportunities to capitalize on. The first, of course, is COSMOS.

We discussed above that COSMOS takes advantage of many desired qualities in a VIRT enabled-information system. It addresses semantics which enables compatibility. Its design is an open architecture that can aid in increasing reliability and affordability. And by using the GIG as a backbone,

[17] Professor Rick Hayes-Roth and Mr. Curtis Blais are currently exploring this area of semantics. In particular, the intent is to develop a common model of *track.* "What does it mean" and "to whom" are a couple of the questions their research is trying to answer.

[18] Governmental systems are generally "stove-piped." They often have difficulty communicating with other systems, are expensive, weight restrictive, and typically live longer than they should due to an inability to accommodate technology advances. COSMOS will use new technology with a focus on information-sharing, all using an open architecture approach.

allowing for interaction with multiple information sources such as CEC, the accuracy of the information is also increased. All of these are desired qualities in an effective coalition information-sharing system. The one it doesn't have is the ability to filter this information based on value. That is, there are no plans to implement VIRT. The perfect opportunity exists to change that. Rather than waiting for COSMOS to be developed and then introducing VIRT, we should do it now. The COSMOS project manager should be briefed on VIRT in order to explore potential implementation of a VIRT service rather than waiting for COSMOS to be fielded without even considering it.

The second opportunity we have is through the Comprehensive Maritime Awareness (CMA) Joint Concept Technology Demonstration (JCTD). The CMA JCTD has two objectives. They are:

1) Demonstrate the value of and implement information exchange and management through employment of a Services Oriented Architecture (SOA) to improve Maritime Domain Awareness (MDA). Relevant maritime activity information will be acquired, integrated, managed and disseminated. Regional threats will be identified using available information. Limited interdiction and inspection assets can then be focused on the most probable threats.

2) Demonstrate net-centric information management for improved MDA, applicable across US Government Departments, Combatant Commands and Coalitions. Integrate with an SOA based on FORCEnet Services Infrastructure (FSI) technology that will provide initial capability to COCOMS (PACOM, NORTHCOM, EUCOM), and have a technology transition path to Program(s) of Record (PORs). Data will be made visible, available, and usable when and where needed. Metadata tags will be implemented to support discovery by users and fusion. Data will be published to an enterprise supporting client subscribers and allowing global data dissemination governed by security and policy controls. (CMA ID, 2005)

VIRT is an obvious candidate for inclusion into this demonstration. Additionally, because the Republic of Singapore (ROS) is a partner in the JCTD we have the potential to show the benefits of VIRT in a joint environment. One of

the main benefits of implementing VIRT in this demonstration is that the scope of the JCTD is much smaller than that of COSMOS. This will likely mean that changes in design architecture and implementation of this new concept can be made more easily and with less bureaucracy.

And lastly, while COSMOS will be open-architecture, it will not be a product-line architecture. A product-line architecture addresses many of the desired qualities mentioned previously. For example, it can reduce cost through reuse and enable compatibility through the use of interchangeable components. The ideal VIRT-enabled coalition information-sharing system will take advantage of this type of architecture. I have designed a notional, product-line architecture for the sharing of information among coalition members. It is called the Advanced Coalition Information Distribution System (ACID). A thorough design, analysis and discussion of this architecture can be found in Appendix A.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. FUTURE RESEARCH OPPORTUNITIES

A. MODELING AND SIMULATION

1. A Robust Model

The area of modeling and simulation of Theory 1 and Theory 2 processes holds the greatest potential for future research opportunities. This thesis merely sets a framework and identifies the pieces that could be modeled in a comprehensive simulation in order to show the value of VIRT. The simple simulation provided in this work needs to be significantly enhanced and developed more fully in order to be able to determine the extent to which a VIRT-enabled information-sharing architecture provides benefit.

The Theory 1 and 2 models provided here lack granularity and refinement. Future research should include a complex set of nodes and with a robust scenario. The following are specific ways that each element in the Theory 1 model could be improved:

- Processing Entity (PE) – For the models simulated in this thesis the processing entities were a single node. In reality, the system cannot truly be tested without several PEs. This is because the real world will include multiple, independent, classes of PEs, all submitting queries. Various ships, aircraft, and coalition participants should all be simulated in a robust warfare scenario as they are likely to place value on different types of information (i.e., have differing COIs).
- Query Specifier (QS) and Query Planner (QP) – These should be two completely different nodes as outlined in Chapter II. For the purposes of modeling in our case we combined them into one node. Whether the Query Specifier will end up lying on the backbone of the GIG or on the ship or aircraft that submits the query should be further examined. The only way to do that is to provide separate nodes for them both. It is my conjecture that the

Query Specifier will likely be another node located on the querying vessel, while the Query Planner ideally will be a service running on the GIG.

- Information Directories (IDs) – As with the QS and QP, the Information Directory (ID) and the Information Store (IS) are distinctly different nodes which, for the purpose of simplifying the modeling, were combined here. Information Directories should be created from the available information stores. In other words, the next step would be to catalog the varying information sources that are available, creating a directory of them. There could easily be more than one Information Directory created, particularly if the directories are split by type of information – e.g., intelligence, METOC, geographical, etc. The creation of a database, or several databases, could likely simulate the Information Directories.
- Information Stores (IS) – When we look at the potential the GIG provides, it can enable every agency, organization, or operational ship or aircraft to contribute an information store. Future research must take this into account and create a way to model the input of information from these varying ISs and make it available to the Information Directories. The intention is not for Information Stores to provide all of their information to the IDs. Rather, the ISs communicate with the specific IDs and provide information as needed. More importantly, the ISs also provide an inventory of the type of information they are capable of sharing. Once again a dynamic database connection might also help to simulate this.

As with Theory 1, Theory 2 holds some room for improvement and further research. Two specific nodes in Theory 2 could be significantly enhanced with further modeling and simulation as follows:

- Condition Specifier (CS) – Rather than using random variables that simulate the exchange of relevant information at varying

intervals, a better measure would be to have a set of conditions that need to be met. This could be done a variety of different ways, but the end result needs to be a simulation that is robust enough to generate varying requests based on a specific set of conditions. These conditions would be modifiable by the user.

- Condition Monitor (CM) – For the simulation here I relied on the assumption that only a very small percent of the information generally requested in a Theory 1 process is actually relevant. (RHR, 2005). What this meant was that model created for Theory 2 assumed that queries generated would elicit a response of much shorter message length. A better and more robust model would modify the Condition Monitor to recognize when information is received that actually matches a set of predefined criteria that is generated via the Condition Specifier (potentially held in a database.)

2. Focus on Value and Semantics

With respect to VIRT I have emphasized two recurring themes throughout this thesis; the *value* to the operator of the information received and the need for a common *meaning* of the information received. Both of these are essential for a VIRT-enabled system.

VIRT relies on the notion that information has value and that some information has more value to one person than another. The way a VIRT-enabled system determines value is a prime topic for future research. How does the system rank order information from different sources? For instance, we have discussed that there will be a myriad of different sources used as Information Stores. All of these will likely have some input into the GIG and ultimately to information used by a particular user on a specific ship, aircraft, or station. How do we determine the quality of that information? In other words, does one information source have better information and therefore, is it more valuable to the user? CEC produces “fire-control-quality” track data. If CEC were to be able to maintain this level of granularity in the data it provides to external users, does

that mean CEC receives a higher value than a coalition ship that is providing data that may or may not have the best sensors? Additionally, information may not always be provided at the same quality, even if it is from the same source. For instance, if a ship's navigation systems go down, the information it provides will likely lose some of its usual quality. After all if a ship or aircraft doesn't know precisely where it is, it cannot provide reliable data. Additionally, *information* means different things to different users so there must be a common framework to ensure that when we request *information*, we get what we want. These questions of semantics merit further research and must be included in any serious study of VIRT. Therefore, the next logical steps for future research with respect to VIRT should include those in the following paragraphs.

A set of metrics must be defined to determine the quality of the information received from an Information Store. These metrics must then be implemented in the model so that we can realistically simulate a value hierarchy of data sources. These can include, for example, the type of navigation system the ship or aircraft uses. A U.S. ship that is using GPS will get a higher value rating as an information source than a coalition ship that is not GPS-equipped. This will allow us to simulate taxing the system in order to determine if it is smart enough to drop the lowest-valued sources first when it becomes bandwidth constrained.

A specific set of mission-impacting parameters must be developed to test the capability of a VIRT system to detect events that match conditions of interest. One potential way to do this is to present various command and post-command level officers with a number of scenarios. The responses generated from the interviews with these officers can be used to develop a database of COIs. These conditions of interest represent the value to a warfighter under those specific circumstances. Using these responses we can measure the effectiveness of VIRT to filter and allow only that information meeting the COIs to flow.

Lastly, a semantic model must be generated to address the issue of interpretation of data. *Track* is a perfect example. (Hayes-Roth, 2005) If you ask five individuals what they consider a track you will get five different answers. To

some it may be a particular contact's latitude and longitude. To others it would include altitude, range and bearing, and course and speed. This presents a significant problem with respect to information-sharing. Development of a model that is able to distinguish these differences in semantics, understands what is being requested, and provides the desired output has to be on the agenda

3. CEC Software Integration

The initial intent of this thesis was to use a Sun Blade server to run a simulation using software provided by PEO IWS. The goal was to develop models in much the same way as was done here, but also to use the provided software for the information source. This information would simulate data as it would be provided by CEC. It would be in the same format and have the same parameters as if we were receiving it from an CEC network. The idea was to actually show how CEC can supply information using a VIRT-enabled process (Theory 2). Some software data were provided, but the binary data format employed made its inclusion in our modeling impractical. The initial software was a binary data file produced by a Cooperative Engagement Processor (CEP) simulator. The data were not received in time to support my research and we had no CEP simulator ourselves to generate such data.

This meant we could not use the CEC simulation data and had to look for another option. We contacted the CEC program personnel at Johns Hopkins University Applied Physics Laboratory (JHU APL) and requested the information in a friendlier format. The result was that the engineers at JHU APL put the information in XML format so that it was easier to integrate. Unfortunately, we did not receive the new data in time for it to be included into this thesis and my simulations. However, the new data may create an opportunity to use actual CEC data to test Theory 2 in subsequent research efforts.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

9/11 Commission Report. Accessed from http://www.9-11commission.gov/report/index.htm, Feb 2, 2006.

O'Rourke, Ronald. "Navy Network-Centric Warfare Concept: Key programs and Issues for Congress." Accessed from http://www.history.navy.mil/library/online/navy_network.htm, Mar 5, 2006.

Air and Marine Operations Center (AMOC) Systems Management Technical Fact Sheet. 2005

Air and Marine Operations Center (AMOC) Operations Fact Sheet. 2005

Balisle, Phil, and Bush, Tom. "CEC Provides Theater Air Dominance." United States Naval Institute Proceedings May 2002: Accessed from http://www.military.com/NewContent/0,13190,NI_Balisle,00.html, Jan 4, 2006.

Busch, Daniel, and Grant, Conrad J. "Co-operative Engagement Capability." Military Technology Aug-Sep. 2003: 26+.

Bush, Daniel, and Grant, Conrad J. "Changing the Face of War: The Cooperative Engagement Capability." Accessed from http://www.ccii.co.za/company/press_releases/face_of_war.html, Feb 9, 2006.

Buss, Arnold. "Simkit Analysis Workbench for Rapid Construction of Modeling And Simulation Components." Naval Postgraduate School, 2004.

Buss, Arnold. "Component Based Simulation Modeling With Simkit." Naval Postgraduate School, 2002.

Buss, Arnold. "Building Complex Models With Legos." Naval Postgraduate School, 2002.

Buss, Arnold. "Modeling With Event Graphs." Naval Postgraduate School, 1996.

Clark, Vern. Sea Power 21. Accessed from http://www.chinfo.navy.mil/navpalib/cno/proceedings.html, Feb 2, 2006.

CJCS. Joint Vision 2020. Accessed from http://www.dtic.mil/jointvision/jv2020b.pdf, Sep 6, 2005.

Cooperative Engagement Capability. Accessed from http://www.whatreallyhappened.com/RANCHO/CRASH/TWA/CEC/cec.html, Jan 17, 2006.

COSMOS ACTD Brief. Defense Information Systems Agency. Accessed from https://www.les.disa.mil/public/COSMOS%20Brief.pdf, Mar 12, 2006

Douangaphaivong, T. Littoral Combat Ship (LCS) Manpower Requirements Analysis. Naval Postgraduate School. Accessed from http://www.nps.edu/Research/HCS/Docs/Douangaphaivong_thesis.pdf, Sep 13, 2005.

Doran, W.F. "Uniting and Securing the Pacific through Technology." Accessed from http://www.chips.navy.mil/archives/03_winter/webpages/pacific.htm, Sep 17, 2005,

Druggan,Tom. "Sensor Network Helps Clear 'Fog of War'." National Defense Magazine. Accessed from http://www.nationaldefensemagazine.org/issues/2001/Jun/Sensor_Network.htm, Mar 1, 2006.

Erwin, Sandra I. "Air Warfare's Holy Grail: 'A Single Integrated Air Picture." National Defense Magazine. Accessed from http://www.nationaldefensemagazine.org/issues/2002/Sep/Air_Warfares.htm, Feb 26, 2006.

Erwin, Sandra I. "Joint 'Air Picture' Fractured By Long-Standing Rivalries." National Defense Magazine. Accessed from http://www.nationaldefensemagazine.org/issues/2001/Dec/Joint%20_Air.htm, Mar 12, 2006.

Fein, Geoff. "Navy To Begin Testing Smaller Cooperative Engagement Capability Systems." Defense Daily Aug 2005: 1.

Hayes-Roth, F. "Model-based Communication Networks and VIRT: Filtering Information by Value to Improve Collaborative Decision-Making." 10th International Command and Control Research and Technology Symposium: The Future of C2, McLean, VA, US Department of Defense, Command and Control Research Program (CCRP), 2005. http://www.dodccrp.org/events/2005/10th/CD/papers/375.pdf

Hayes-Roth, Rick. Towards a Rich Semantic Model of Track: Essential Foundation for Information Sharing. Naval Postgraduate School, 2005

Hayes-Roth, F. "Two Theories of Process Design for Information Superiority: *Smart Pull vs. Smart Push*." Command and Control Research and Technology Symposium: The State of the Art and the State of the Practice, San Diego, CA, US Department of Defense, Command and Control Research Program (CCRP). To appear 2006.

Introduction to Discrete Event Simulation. Accessed from http://www.dmem.strath.ac.uk/~pball/simulation/simulate.html, Feb 3, 2006.

Introduction: EXOCET Family. Accessed from http://www.mbda.net/site/FO/scripts/siteFO_contenu.php?lang=EN&noeu_id=108, Feb 23, 2006.

Kelton, W. David. and Law, Averill M. Simulation Modeling And Analysis. 3rd ed. Boston, MA: McGraw Hill, 2000.

Kerno, Robert. "CEC and the Interoperability Challenge." Accessed from http://www.navyleague.org/seapower/cec_and_the_interoperability_cha.htm, Jan 23, 2006.

Lambie, Chris. "System Would Give Yanks Control of Canadian Missiles." Daily News [Halifax]. Accessed from http://www.polarisinstitute.org/polaris_project/corp_security_state/Opinions_interviews/january_15_2004.htm, Feb 13, 2006.

Lundquist, Edward. "Cooperative Engagement Capability Provides Combat Capability Through Connectivity." Naval Forces 2005: 56+.

McKenna, Ted. "US Navy CEC Program Changes Tack" Journal of Electronic Defense Mar 2004: 39.

Modeling & Simulation: An Introduction. Accessed from http://www.systems-thinking.org/modsim/modsim.htm, Mar 5, 2006.

Mullen, Mike. Sea Enterprise: Resourcing Tomorrow's Fleet. Accessed from http://www.usni.org/proceedings/Articles04/PRO01mullen.htm, Feb 26, 2006.

Naval Studies Board. The Navy's Needs In Space for Providing Future Capabilities (2005). Accessed from http://newton.nap.edu/books/0309096774/html/123.html, Mar 5, 2006.

Oros, Carl. A Proposed Architecture for a Helicopter Information Awareness Model (I-AM). Naval Postgraduate School, 2004.

Persian Gulf Wars. Accessed from http://www.infoplease.com/ce6/history/A0838511.html, Mar 1, 2006.

Pierce, Terry C. "Sunk Costs Sink Innovation." United States Naval Institute Proceedings May 2002: Accessed from http://www.military.com/NewContent/0,13190,NI_Sunk,00.html, Jan 7, 2006.

Rivers, Brendan P. "Victory At CEC." Journal of Electronic Defense Sep 2001: 40-46.

Software Product Lines. Accessed from http://www.sei.cmu.edu/productlines/index.html, Mar 6, 2006.

To improve Joint Interoperability, Navy restructures its CEC Program. Accessed from http://www.globalsecurity.org/military/library/news/2004/01/mil-040130-navsea02.htm, Mar 5, 2006.

Walsh, Ed. "Surface Navy Transforming CEC for Current, Future Combatants." United States Naval Institute. Proceedings Jun 2005: 91-92.

APPENDIX A. ADVANCED COALITION INFORMATION DISTRIBUTION SYSTEM (ACID)

A. OVERVIEW

This appendix provides a notional system for a VIRT-enabled coalition information sharing system of the future. The ACID system uses as its construct the concept of a product-line, component architecture to implement VIRT. In order to understand the benefits obtained by adopting a product-line architecture lets delve into exactly what a product-line architecture is.

B. PRODUCT-LINE ARCHITECTURE

What is a product-line Architecture (PLA)? Perhaps the best definition of a PLA comes from Carnegie Mellon's Software Engineering Institute (SEI). They define it as follows:

> A software product-line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way. (SEI, 2006)

The concept of product-line architectures is not a new one. In fact, it is a safe bet that almost every one reading this thesis has had first hand experience with one major system that was built using a product-line architecture. I am of course referring to the automobile. There are many different kinds of cars and a multitude of car manufacturers. We are talking dozens of different "products." General Motors, Ford, Daimler Chrysler and BMW are just a few of the automakers currently in business. Additionally, each of these corporations manufactures a number of different models of vehicles. For GM there is the Tahoe, GTO, Impala, and Corvette just to name a few. Ford has the Mustang, Five Hundred, GT, and Focus. And Daimler-Chrysler has the Charger, Ram, 300, and Durango. Although it can be argued that these vehicles do not look the same or even perform the same functions as each other, it is extremely likely that each shares a large percentage of their parts and components with the others. That is the true advantage of a product-line architecture.

Let’s take a closer look at how this advantage is realized. A typical car will contain thousands of parts. If an automotive manufacturer were to retool the assembly line and manufacture all new parts every time they brought a new car to market it would be a very expensive venture. A business has only one purpose, to make money. Yes, the organization may have many stated goals or purposes, but in the end it is profit that drives a firm’s actions. Instead, although an automobile has a multitude of parts, it has a much smaller set of distinct modules that make up the car. For example, each car has a drive train, some type of steering setup, and a braking system. The manufacturer uses these modules, as shown in Figure 18, to create the cars we all drive. What makes the modules (or product-line) architecture different than the example in which all the

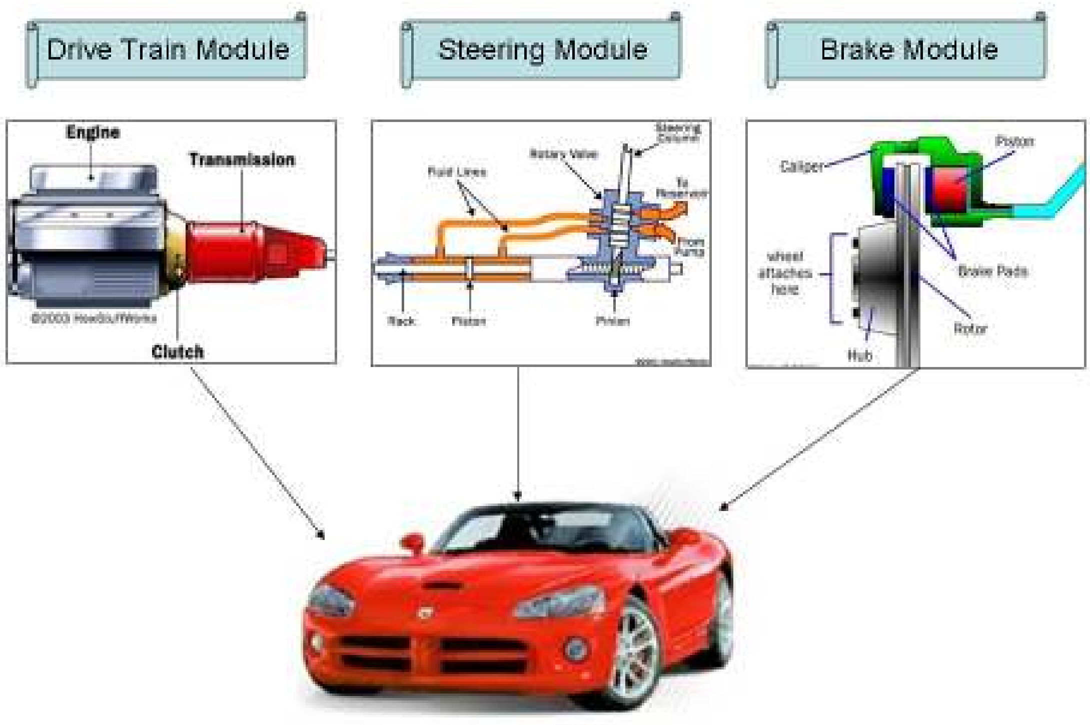

Figure 18. Automotive Product-line Modules

parts are recreated is the idea of reuse. Reuse is absolutely essential for both minimizing cost and rapidly deploying a new system instead of having to start from scratch. In our car example, the Dodge hemi engine is currently being used

in six vehicles including the Charger, Ram, Durango, 300C, Crossfire, and Magnum. The engines may have to be modified in order to ensure they work correctly with the components of the rest of the car, but in most cases modification is quicker, easier, and cheaper than beginning anew.

Although the car analogy discusses primarily the idea of hardware reuse, the same trend has been on the rise in the software community. The primary reason is that technology is growing at a rate so fast that it is extremely difficult to keep up with it. Creating programs instead of modifying and reusing programs (or parts of programs) wastes valuable time. In addition, the excessive cost, difficulty, and time required to create new systems from scratch has become prohibitive. Product-line architecture can solve that problem. It only makes sense that as we look to VIRT-enabled future systems that we adopt a product-line architecture to do so. This is what I have done with the ACID system. Once again it is important to understand that a product-line architecture is not a system, but rather a design process or framework for many systems using a set of similar components but often having varied functional requirements.

C. DESIRED SYSTEM QUALITIES

We have talked about the fact that the United States and its coalition partners have no efficient or effective means of communicating and sharing information. We have discussed COSMOS and its shortfalls. That is where ACID comes in. ACID is a framework for a VIRT-enabled, product-line architecture that accomplishes the goals of both COSMOS and VIRT. In this case, let's share the right information amongst all participants (coalition, CEC, AMOC, etc) at the right time. In addition, this framework ensures a system meets some specific requirements and has particular attributes. These desired qualities include getting information to the user quickly (low latency), being fully compatible with all partners, is extremely reliable, and both accurate and affordable. The ability of the operators to get the high value information they want quickly and without the added low value chaff will significantly impact the ability to accomplish assigned Joint missions. Imagine the user having access to an information-sharing system that was designed from the beginning with all of

these characteristics. ACID provides a foundation for the following desired capabilities:

1. Quick (Low Latency)

There is an expression "Speed Kills." In military operations a similar phrase is germane and it is "Lack of Speed Kills." The pace of military operations can be extremely taxing both on personnel and systems. Decisions must often be made in seconds or minutes rather than hours or days. That means that the decision support systems and data sources that feed and influence those decisions must be timely. They must be updated very quickly with minimal lag time or latency. Coalition partners provide important information for decision making.

2. Compatible

> Interoperability is the foundation of effective joint, multinational, and interagency operations… The future joint force will have the embedded technologies and adaptive organizational structures that will allow trained and experienced people to develop compatible processes and procedures, engage in collaborative planning, and adapt as necessary to specific crisis situations. These features are not only vital to the joint force, but to multinational and interagency operations as well. (*JV2020*)

It is important today more than ever that we be able to integrate and operate seamlessly in a Joint environment. That means amongst services, across agencies, and throughout a multi-national environment. It is insufficient to assume, as the Joint Technical Architecture does, that simply requiring adherence to physical standards will ensure systems are compatible.[19] More importantly, that architecture only applies to US systems and as such does little to ensure our interoperability with foreign vessels/agencies. As mentioned above, the number of countries we operate with keeps increasing. This further exacerbates the issue of compatibility. It is simply politically incorrect and unacceptable to be able to share information with one coalition partner in an exercise and not be able to share that information with a different coalition

[19] The Joint Technical Architecture is a good first step in that it helps to ensure that all future Governmental systems operate using hardware based on a set of defined standards. This will aid in interoperability, but will in no way ensure it. https://disronline.disa.mil/a/DISR/docs/jta-vol-I.pdf.

member. If CEC is going to share information effectively, compatibility amongst all partners must be addressed

3. Reliable

Systems are only as good as they are available. Once they go down or become inaccessible, they immediately begin to lose their value, and consequently, can have significant military impacts. Although every military leader would like his or her decision support and information systems available 100% of the time, this is not a realistic goal. Things will go wrong, and systems will go down. This means the system must be capable of synching information from/to all sources or participating units periodically so the system will remain operational in case of an outage of the Information Vault (the source of the information being provided). Additionally, it must be capable of re-synching with the correct information once the Information Vault comes back online. Because of the architecture's reliance on the GIG and other global information systems (coalition networks, GPS, etc.), providing a completely redundant Information Vault in case of an outage is not a viable option.[20]

4. Accurate

Data means nothing if you cannot depend on its validity. It can be abundant, timely, and always available, but will be useless if an operator can't trust it. That means that the system must consistently provide not just data, but the correct data to the right operator. It also means the problem of "dual tracks"[21] must be solved in the new system. After all, a common operational picture (COP) must be just that – common. That means that all stations must hold the most accurate, reliable and relevant information available. Additionally, that information should be correlated so that each ship and aircraft has as near the same picture as possible. This is one of the founding principles of CEC and

---

[20] A primary benefit of using the GIG as a backbone is that it is like the Internet in the sense that the possibility of the entire network going down is low. However, the system on the ship is much more likely to experience outages.

[21] Dual tracks are those instances where a system receives data from one or more feeds that disagree with the data it holds organically. The system, not correlating the foreign radar video to the one it currently holds on its own sensors, inserts a duplicate contact. While these contacts are indeed the same, the operator can become confused and subsequently error is introduced into the decision-making process.

is what makes CEC data so valuable. Just as with CEC, the coalition system must ensure that all track data is fused and synched allowing all vessels to have a truly common operational picture.

5. Affordable

Cost will always be an issue. Its impact is magnified significantly when you inject the notion of multi-national partners. It is impossible to attain interoperability with coalition members if the system required is more expensive than they are willing or able to commit to. Yet, make no mistake, we must operate multilaterally.

> In all cases, effective command and control is the primary means of successfully extending the joint vision to multinational operations. Technological developments that connect the information systems of partners will provide the links that lead to a common relevant operational picture and improve command and control. (*JV2020*)

We must ensure that we don't assume that all other countries that will need to use the ACID system will have the money to pour into it that we traditionally associate with Navy acquisition. By employing a product-line architecture we can significantly reduce cost of supporting diverse users and increase the number of coalition partners we eventually become compatible with. That is one of the reasons COSMOS is using an open architecture, service oriented framework.

D. THE ACID SYSTEM FRAMEWORK

1. Functionality

In order for the ACID system to be of any use, it must have some core functionality. That is, there are certain essential functions that are required if the system is to be successful.

a. Real-Time Sharing

Working with both our own forces and coalition forces requires us to be speaking the same tactical language. This is difficult in a dynamic environment such as combat or Military Operations Other Than War (MOOTW). In order to coordinate our actions and perform as a cohesive team we must be all be using the same world model. "The key capability required to enable virtual

organizations to coordinate and execute at maximum effectiveness in dynamic environments is a *shared world model.*" (RHR, Model-based Comms and VIRT)

b. Bit Prioritization and Bandwidth Maximization

For reasons discussed earlier in this paper, bandwidth is a primary concern. The ACID system will use VIRT to prioritize information flows. By assigning value to specific information from different sources it is possible to maximize the bandwidth by allowing only that information required by the user to flow and then only at the appropriate time. This is Smart-Push. The CM (Condition Monitor) knows what the user needs and what information is valuable to the user *vis-à-vis* mission accomplishment.

c. Fully Autonomous

The system will be fully automated. It will be a set and forget type of system. The set refers to the user's ability to enter parameters for the information prioritization. This can be thought of making the initial query in Theory 2. Once done, the user will have the information provided in the required format and be alerted when changes occur that affect the mission in a negative way or for that matter any way that requires some sort of action by the user. By requiring little to no allocation of personnel resources for this process, the sailor is free to concentrate on other matters of higher priority – including possible enemy engagements.

2. Components

The following components, illustrated in Figure 19, comprise the ACID system:

a. Onboard Sensor Interface (OSI)

The Onboard Sensor Interface is an organic compiler and collector of own ship's data systems and information sources. Sources of input for the OSI include any organic Mine Warfare (MIW) capability such as sleds or sonar systems, air search tracking and acquisition radars, Surface search radars, primary Undersea Warfare Systems (USW) such as bow mounted sonar or passive arrays, GPS receivers, atmospheric sensors (wind, precipitation,

temperature, etc), and Electronic Warfare (EW) equipment. The OSI interfaces directly with the Information Vault via the Domain Translator.

b. Domain Translator

The Domain Translator serves an important purpose. That purpose is to ensure that information is recognizable by the information vault. Ideally, and in the future possibly, both we and our allies will be using the same systems or at least systems based on the same ontology[22]. That is to say that they will be able to understand and recognize the meaning of the information that is flowing from

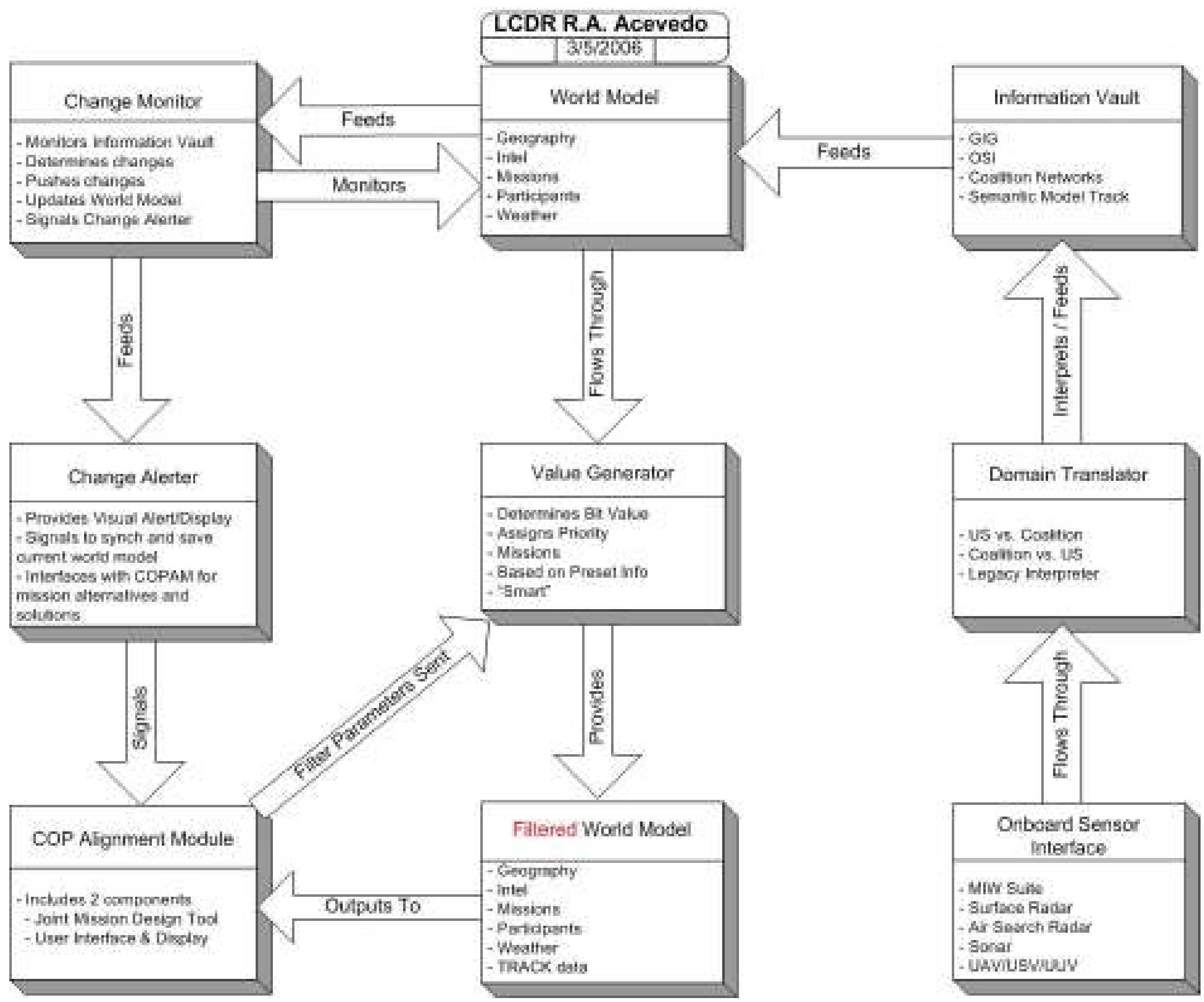

Figure 19. ACID System Framework

22 Ontology is simply a collection of terms or semantics. These terms have been agreed upon and are accepted by those within a particular community of interest. By using systems designed around the same ontology we enable information-sharing. (RHR, 2005)

one unit to another. That will engender true interoperability. In the meantime, there will be legacy systems both in the US and coalition fleets. Additionally, even US to US and coalition to coalition information-sharing will likely require the use of the translator because every country has different ships, each with its own set of stove-piped systems onboard. Future implementations will likely rely on some type of metadata tagging such as XML to standardize data format. New systems will take advantage of this. However, the legacy systems must still be incorporated and is the reason for the translator.

c. Information Vault

The information vault will use as its primary backbone the GIG. However, we must also consider other information sources that may not be flowing to the GIG. Coalition members will also be receiving information from other sources that are not input into the GIG. If CEC and other agencies want this information we must figure out a way to share it. Information-sharing is a two-way street. The Information Vault will be the repository of all of the information input from a variety of sources and will serve to feed/form the World Model. The Information Vault may very well be the GIG, but this framework does not make that assumption.

d. World Model

The World Model can be viewed as a virtual component. The reason is that it is really the result of a collation of all of the data within the Information Vault, scrubbed to provide a one-stop shopping place of all required information. This is the traditional Common Operational Picture we have come to understand. While there are some differences between a COP as we know it and a World Model, they are very similar. A World Model is a set of objects or dynamic models that can perform some inferences autonomously, such as projecting the future when asked or dead-reckoning, as when disconnected from others for some time. (Hayes-Roth, 2005)

e. Value Generator

The Value Generator provides the nuts and bolts of the system. VIRT, after all, dictates that only information deemed necessary should flow.

Additionally, that information should flow without being asked to and in advance of its eventual recipient even perceiving its existence and, thus, without knowing that it's needed. The value generator is what enables that functionality. It labels different types of information based on a user-defined set of parameters. The value generator can also work automatically. Merely give a particular mission type or expected threat level and it will filter data according to a preset selection of rules. Additionally, the value generator is "smart." That is, it can learn and apply what it has learned to future value assignments. This ultimately reduces the load on the operator and prevents unintended mistakes related to parameter entry. This is no small task and will require that the value generator realize that different operators will have a different focus and hence value the same information differently.

f. Filtered World Model

The Filtered World Model is different than what we know as the Common Operational Picture. It would be more appropriate to think of it as the CROP or Common Relevant Operational Picture. It differs from the World Model in that the user sees and has available precisely what he needs--no more, and no less. This ensures the best use of both personnel and communications bandwidth. This model will be different depending on where in the food chain (chain of command, organization, etc.) the individual is. PFC Snuffy[23], for example, probably does not care what other operations are going on throughout Iraq in other towns. He also is likely not interested in what is going on in another part of the same town he is patrolling. He is definitely concerned with, however, the building he is about to enter. That should be his focus, and that of his particular filtered world model.

g. COP Alignment Module (COPAM)

The COPAM has two parts. The first part is the Joint Mission Design Tool. The Joint Mission Design Tool (JMDT) is the primary means for entering parametric information into the system. What is the mission? What are we concerned with? What information do we want to see? All of this is enabled

[23] PFC Snuffy is a fictional individual introduced by Professor Rick Hayes-Roth. The term PFC Snuffy represents a soldier or Marine at the low end of the operational chain of command.

through the JMDT. The JMDT interfaces with the value generator to allow prioritization of bits based on these inputs. Additionally, the JMDT provides recommendations and alternatives that allow for mission accomplishment in light of changes that have occurred to the world model. The JMDT is the component that provides automated alternatives to allow for continued mission accomplishment when notified by the Change Alerter of mission impacting data.

The second part of the COPAM is the User Interface and Display system. This is the actual output of all of the data. Whether it is a 3D or 4D rendering of the battle space on several large screen television sets, this is what the operator actually sees. The operator can also manipulate the way in which the information is displayed. This can include changing symbology, color, etc.

h. Change Monitor[24]

The Change Monitor is responsible for monitoring the world model to determine if there have been changes that impact the set parameters of the mission. A good example would be a surface action group (SAG) transiting the Atlantic Ocean enroute the Mediterranean Sea. The group is due to refuel with a US Navy oiler in 48 hours. Part of the information available in the GIG is the operational schedules of all ships and aircraft. Suppose the oiler that is assigned to the SAG loses two of her engines and is unable to get underway to rendezvous with the SAG. This information is recognized by the Change Monitor as being significant to the mission. That is, it correctly surmises that because the oiler will not be able to get underway and rendezvous with the SAG, the SAG will run out of fuel. It immediately signals the change alerter.

i. Change Alerter[25]

The Change Alerter, having received a signal from the Change Monitor in the above example that the ships will run out of fuel, sends a visual alert (screen blinks) and audible alert (alarm buzzes) to the COPAM to notify the operator. In response to this alert, the COPAM (and the JMDT in particular)

---

[24] Component introduced by Professor Rick Hayes-Roth in *Model Based Communication Networks and VIRT*

[25] Component introduced by Professor Rick Hayes-Roth in *Model Based Communication Networks and VIRT*.

develops and prioritizes up to three alternatives to respond to the change that occurred. In this case the system makes the following analysis and recommendations in order of desirability:

1. Divert to the Azores: Adds three days to the mission
2. Rendezvous with alternate oiler in different location: Adds five days to the mission
3. Divert to Bermuda: Adds seven days to the mission

Additionally, the change alerter signals the backup drive to synch the current world model. This ensures the system has captured any changes and also provides a backup in case of information system outage – such as the GIG. These backups can also be done on a time interval preset by the operator.

3. Quality Attributes

a. Latency

Latency in this case refers to the ability of the system to process the information from all sources and adapt the mission plan to significant events or changes that have occurred in sufficient time to allow for the successful completion of the plan. In the refueling example provided above, the system had to be able to recognize that the broken oiler had an impact on our refueling plan, notify us of the issue, and then provide alternative recommendations in a timely manner so that we would still be able to carry out the mission.

b. Availability

Availability means that the system continually performs as designed in sufficient manner to ensure mission success. If for example, information is flowing, but the change alert module ceases to function, then the system cannot be said to be available. Additionally, because outages will inevitably occur, when the system does again become available it must be able to process the newly available and updated information in a manner that makes sense. Such a manner would include updating the information in a prioritized fashion based on the value of the information or its impact to the mission (ideally the same, but not

necessarily so) instead of simply updating in the chronological order in which it occurred.

c. Accuracy

Accuracy is an important attribute and as discussed here means that the system provides not just the required information, but that the information provided also meets pre-defined required mission parameters. The data received that justifies the adaptive response must be credible and the system must be confident to a degree of certainty of the accuracy of the data and data source.

d. Value

VIRT works through a prioritization of data. Value in the context used here relates to the importance of the information to the user. Additionally, different users and, more importantly, different missions will change the importance placed on a particular bit or piece of information. A good example is that an operator may desire much information to ensure he or she has the most accurate view of the environment as possible. However, not all of this information is essential to a particular mission's success. High value bits would be those that have a direct bearing or impact on the mission. Lower value bits would be those that did not impact the mission but were still desirable by the operator. The quality sought after here with respect to value is that the system not only understands what information is significant to the mission, but is also able to prioritize the data so that the MOST important piece of information flows first. It is equally important that in the event of system degradation the system drops the least important bits first.

4. Functional Requirements

The primary purpose of this system is to reduce bandwidth resource consumption while at the same time providing maximum interoperability and information-sharing both between US vessels and amongst our coalition partners. There are a number of functional requirements that will enable this system, but three core requirements will give the system its heartbeat.

a. Real-Time Data Sharing

For this system to be successful it will be required to provide real-time sharing of information. Significant delays or lag times will render the system useless. The best way to ensure low latency is to only ship those bits that are required. As stated above, the system must send the important bits first. Then if there is excess capacity of bandwidth the low bits can be sent. This will ensure that extraneous (not required) information will not hog bandwidth and slow down the system.

b. Update and Filter World Model with Visual Display of Relevant Information

This is the "Ability to dynamically filter in real time the views, processes, simulations, and predictions of the world model to address the current mission parameters."[26]

c. Redundant Storage with Fully Automated World Model Synch Capability

Because system downtimes are not just plausible but inevitable, it is essential that all units continue to operate using the latest World Model available. In order to do that, redundant storage must be available via a hard drive or other removable media. Additionally, because information flow will be dynamic and outages will be unexpected, the system must continuously synch data from the Information Vault to ensure an accurate World Model. It is important to note that each operator will have a different filtered world model. This is because different operators will require different pieces of the true (Global) World Model.

E. ARCHITECTURE EVALUATION

1. Scenarios

Three scenarios were chosen to evaluate the architecture in a notional environment in order to gain insight into any potential issues that might arise. By providing a product-line architecture, this system can be customized and

[26] This is an excerpt taken from Maj Carl Oros' architecture for a Helicopter Information Awareness Module. The statement is just as pertinent in this architecture and is thus quoted.

modified to meet a wide variety of country/platform/use requirements. The scenarios are as follows.

a. Open Ocean Surface Action Group (SAG) Steaming

This scenario simulates a group of 2 or more surface vessels in an open ocean steaming environment. The number of contacts encountered is limited and the complexity of the environment is low.

b. Low Slow Flier

This is a fairly robust scenario in that it involves a helicopter flying very low to the surface and very slowly. This could be the result of terrorist activity or a simple oil platform helicopter. The complexity is medium to high as the system will likely interpret the helicopter initially as a surface track and route the track info to the Surface Tracker and NOT the Air Tracker. Once the item is recognized as an air track, through an increase in altitude or velocity, the system will have to alert the appropriate operator and pass the data to the Air Tracker automatically.

c. Refueling Mission where AOE is Out of Commission

Ships at sea are required to refuel at sea. They accomplish this by rendezvousing with an oiler, or AOE, and then conducting a transfer of fuel. This scenario is of low to medium complexity because it simply requires the system to respond to the fact that the oiler scheduled to refuel the SAG has broken and is no longer available. The system must alert the user to this fact and then calculate and recommend alternatives that allow for mission accomplishment. This scenario could have been made more difficult (although architecturally the same), if the US SAG had been originally scheduled to be refueled by an Australian oiler.

2. Sensitivities

As expected, these scenarios did identify certain sensitivities. They are as follows:

a. Mission Updating and Rerouting

The system can be set to synch at any given interval. The issue is that the system can only plan alternate routes based on its current world model.

That means that if it is operating with an imprecise world model (or out-of-date copy), the alternatives will not be accurate and may not even be viable. Remember, the primary purpose for the synch is merely as a measure of redundancy. The World Model (and Filtered World Model) will be continually changing and updated based on the information flowing to and from the Information Vault. By synching the system you are in essence freezing and saving to HDD the current World Model. This copy will act as a file backup should the ship or aircraft lose connectivity. In the case of an outage, the system will automatically revert to the latest synched copy, thus assuring the system is able to continue operating. Additionally, the system will automatically synch the high priority information the instant it detects an outage.

b. System Latency and Synch

Synching the world model has the potential to have a significant impact on latency. The desired parameter would be to have the system synch at intervals less than 1 second. This is particularly important with the prospect of target engagement. The issue is the more often the system is synched, the more latency is introduced. One way to mitigate this would be to synch only that information with high value. This would mean that only those bits that have direct impact on mission accomplishment would be synched and saved. The lower value bits (those that don't directly impact the mission but are still desirable by the operator) could be synched as well, but at a much lower frequency.

c. Classification of Data

Our data classification system currently consists of access or clearance and need to know. Need to know must be considered inherent for the information in the system to flow. Additionally, information flow is routed by station or operator by design and not by individual. This is particularly critical when sharing data between US and coalition partners. Having an emphasis on security can significantly increase latency. There will likely be a tradeoff between latency and security. The GIG is envisioned as a system of services. The notion is that it will be a backbone with many services running on it. Because the GIG portends to be the portal through which the majority of information-sharing will

occur, I anticipate that there will have to be some service running on it that acts as a filter for various classifications of data.

d. Value of Data

The ability of the system to generate a proper value for each bit of information is essential to its operation. Without it functioning properly, the operator will likely be overwhelmed, and the communications pipe clogged. The sensitivity here is that if the system assigns the wrong value to the data, the operator may never see it. This is primarily an issue when the system assigns a lower than should be value to the bits rather than a higher than should be value. One potential way around this would be to have the system "learn" through use. In other words, the more the system filters the bits correctly, the better and more reliable it will become. It would work similar to a vehicle navigation system. Current car navigation systems use a Global Positioning Receiver mounted to the vehicle to determine its position. Through routine use over time, the system gets better at telling the driver when to make the turns correctly. The same is true for the ACID system. The more it filters, the better it can become.

3. Tradeoff Points

As with the implementation of most information systems, there are tradeoffs. This system is no different.

As discussed earlier, synch frequency can have a dramatic impact on latency. Without the synch, however, the system would be rendered useless during an outage of the Information Vault or its inputs. A balance must therefore be struck between the two. Because these parameters are designed to be adjustable, it is likely that the best combination of latency and synch will be achieved through system use and evaluation by the requisite operators.

Value is the root of the second tradeoff. There is a point where the operator will have too much information and be overloaded. In warfare, however, having too little information when it is really needed can be devastating and cause loss of life. It is likely that at least initially, operators will assume they are not getting the information they want and adjust the requested information so that they receive almost everything. The goal therefore is that the value generator be

very adept at determining, based on user input and past situation learning, the accurate measure of information value. This will give the user more confidence in the system with use over time. That said, as with the synch feature, use of the system will determine the appropriate measure of value and whether the system is really doing its job.

The final tradeoff will be between latency and security. As previously discussed, high levels of security imply high latency. In order for most systems (including this one) to be effective, the system must have low latency. Therefore, there must be a tradeoff that will give the flexibility to share the required data while still providing an acceptable level of security.

4. Risk Management Strategy

Study of the scenarios has yielded a number of potential pitfalls and hence raised some issues with respect to mitigation. The following section attempts to illustrate these risks and their corresponding prevention measures.

a. Loss of Information Vault

The system hinges on a valid world model that is common across the operational domain. Whether that be a Carrier Strike Group, Expeditionary Strike Group, or Joint Task Force, it is essential that everyone be on the same page. The Achilles heel is the Information Vault. It is the crux of information flow that enables the world model. It will, at times, go down. It is inevitable. With that in mind, the HDD backup and synch features assure that the latest world model is always available. This gives the system the ability to run disconnected, an attribute that is essential.

b. Value Assignment

Value assignment of the information is yet another potential risk. If it works incorrectly then the operator could either see too much information or not enough. It is for this reason that the system also allows the users to modify the desired value parameters in order to ensure that the information they are getting is what they want.

c. Classification of Data

This is a risk, particularly when working with coalition ships. Ensuring that the system will only allow that information for which the coalition members are cleared to receive will be a challenge. That said the system can filter ANY information required. This should enable the operator to manually block the data that is deemed classified. The inherent problem in this is that the system should be as automated as possible in this regard. The more human interaction, the more time spent not doing critical functions of the job and wasting time. This additionally further introduces possible error. Hence the reason the most appropriate solution is likely to be in the form of the classification filter service that should run on the GIG. The biggest benefit is that ALL systems, and not just the ACID system, can then benefit from the classification filtering capabilities.

d. Filtered World Model and Alternative Plans

As designed, the system will identify changes to pertinent information and alert the user. It will then make recommendations for alternative candidate plans. The instance may arise when the one of the system's alternatives needs a piece of information not provided by the filtered world model. This would result in an alternative not being available. To mitigate this, the Change Monitor monitors the World Model and not the Filtered World Model in order to ensure it has access to the appropriate information in the event of a change. Additionally, as with most other information systems, the operator has override ability. This means that if the operator doesn't have the information they need, or think they are getting a plan based on incomplete data, they can go get the information they desire.

F. ACID AND CEC

When warfighters make decisions they want all of the relevant and significant information available to help them make that decision. Sometimes a single piece of information can mean the difference between taking action that kills many people and failing to take action that similarly results in deaths. CEC was developed on that premise. However, CEC provides just a single piece of

information. Our coalition members have additional information and are an important part of ensuring we have what we need to make those informed decisions. We must be able to operate with them and share information seamlessly in order to accomplish this. CEC equipped ships have a very valuable information sharing system. As discussed earlier, it is perhaps the best air tracking system available. Coalition members would benefit greatly from that data. COSMOS is a good start, the ACID framework I propose here, while notional, represents an ideal solution. ACID will enable CEC equipped aircraft and surface combatants to share information efficiently with our allies. In addition, the component architecture means that the longevity of the system will be significantly increased. CEC is one part of the puzzle and ACID is another. VIRT is the glue that binds them and creates an effective system.

APPENDIX B. SMART PULL CODE

A. CODE OVERVIEW

The code contained in Appendix B was created by Curtis Blais of the NPS Modeling, Virtual Environments, and Simulation (MOVES) Institute for the implementation of the model described in Chapter IV. This code represents the Theory 1 model. The code and parameters contained herein may be used as required to further instantiate future simulation models of the VIRT concept.

```
/*
 * SmartPull.java
 *
 * Created on February 7, 2006, 12:00 PM
 */

package virt;

import simkit.*;
import simkit.random.*;
import simkit.stat.*;
import java.util.*;
import java.text.*;

public class SmartPull extends SimEntityBase {

   private double queryTime;
   private int queryLength;
   private int queryInterval;
   private double responseTime;
   private long responseLength;
   private long transmissionSpeed;
   private double processingTime;

   private int queriesWaiting;
   private int responsesWaiting;

   protected long bandwidth;
   protected int userPE;

   /** Creates a new instance of SmartPull */

   public SmartPull(double queryTime,
         int queryLength,
         int queryInterval,
         double responseTime,
         long responseLength,
         long transmissionSpeed,
         double processingTime) {

      setQueryTime(queryTime);
```

```
    setResponseLength(responseLength);
    setQueryLength(queryLength);
    setTransmissionSpeed(transmissionSpeed);
    setQueryInterval(queryInterval);
    setProcessingTime(processingTime);
    setResponseTime(responseTime);
}

/** Set initial values of all state variables */
public void reset() {

    super.reset();

    /** StateTransitions for the Run Event */

}

public void doRun() {
   userPE = 1;
   firePropertyChange("userPE", getUserPE());

   queriesWaiting = 0;
   responsesWaiting = 0;

   bandwidth = 1; //network resource
   firePropertyChange("bandwidth", getBandwidth());

   waitDelay("InitiateQuery",0.0,new Object[]{},0.0);

}

public void doInitiateQuery() {
    /* Code insertion for Event InitiateQuery */

    /* End Code insertion */

    if (userPE > 0) {
        waitDelay("StartQueryProcessing",0.0,new Object[]{},0);
    }
    else {
        queriesWaiting += 1;
    }
    waitDelay("InitiateQuery",queryInterval,new Object[]{},0);
}

public void doSendQuery() {
    /* Code insertion for Event SendQuery */

    /* End Code insertion */
    /* StateTransition for bandwidth */
    long _old_Bandwidth = getBandwidth();
    bandwidth = bandwidth - 1;
    firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());

    /* StateTransition for userPE */
    int _old_UserPE = getUserPE();
```

```
      userPE = userPE + 1;
      firePropertyChange("userPE", _old_UserPE, getUserPE());

      checkWaiting();

      if (true) {
         waitDelay("NetworkReceiveQuery",(double)queryLength/(double)transmissionSpeed,new
Object[]{},0);
      }
   }

   public void doNetworkReceiveQuery() {
      /* Code insertion for Event networkReceiveQuery */

      /* End Code insertion */
      /* StateTransition for bandwidth */
      long _old_Bandwidth = getBandwidth();
      bandwidth = bandwidth + 1;
      firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());


      waitDelay("NetworkSendResponse",responseTime,new Object[]{},0.0);
   }

   public void doNetworkSendResponse() {
      /* Code insertion for Event NetworkSendResponse */

      /* End Code insertion */
      /* StateTransition for bandwidth */
      long _old_Bandwidth = getBandwidth();
      bandwidth = bandwidth - 1;
      firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());


      if (true) {
         waitDelay("ReceiveResponse",(double)responseLength/(double)transmissionSpeed,new
Object[]{},0);
      }
   }

   public void doReceiveResponse() {
      /* Code insertion for Event ReceiveResponse */

      /* End Code insertion */
      /* StateTransition for bandwidth */
      long _old_Bandwidth = getBandwidth();
      bandwidth = bandwidth + 1;
      firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());


      if (userPE > 0) {
         waitDelay("StartProcessingResponse",0.0,new Object[]{},0);
      }
      else {
         responsesWaiting += 1;
      }
```

```
    }

    public void doStartProcessingResponse() {
        /* Code insertion for Event StartProcessingResponse */

        /* End Code insertion */
        /* StateTransition for userPE */
        int _old_UserPE = getUserPE();
        userPE = userPE - 1;
        firePropertyChange("userPE", _old_UserPE, getUserPE());


        if (true) {
            waitDelay("EndResponseProcessing",getProcessingTime()*responseLength,new
Object[]{},0);
        }
    }

    public void doEndResponseProcessing() {
        /* Code insertion for Event EndResponseProcessing */

        /* End Code insertion */
        /* StateTransition for userPE */
        int _old_UserPE = getUserPE();
        userPE = userPE + 1;

        checkWaiting();

        firePropertyChange("userPE", _old_UserPE, getUserPE());


    }

    public void doStartQueryProcessing() {
        /* Code insertion for Event StartQueryProcessing */

        /* End Code insertion */
        /* StateTransition for userPE */
        int _old_UserPE = getUserPE();
        userPE = userPE - 1;
        firePropertyChange("userPE", _old_UserPE, getUserPE());


        if (true) {
            waitDelay("SendQuery",queryTime,new Object[]{},0);
        }
    }

    public void checkWaiting() {
        if (queriesWaiting > 0) {
            waitDelay("StartQueryProcessing",0.0,new Object[]{},0);
            queriesWaiting -= 1;
        }
        else if (responsesWaiting > 0) {
            waitDelay("StartProcessingResponse",0.0,new Object[]{},0);
            responsesWaiting -= 1;
```

```
    }
  }

  public void setQueryTime(double queryTime) {
    this.queryTime = queryTime;
  }

  public double getQueryTime() {
    return queryTime;
  }

  public double getProcessingTime() {
    return processingTime;
  }

  public void setResponseLength(long responseLength) {
    this.responseLength = responseLength;
  }

  public void setResponseTime(double responseTime) {
    this.responseTime = responseTime;
  }

  public void setQueryInterval(int queryInterval) {
    this.queryInterval = queryInterval;
  }

  public void setProcessingTime(double processingTime) {
    this.processingTime = processingTime;
  }

  public long getResponseLength() {
    return responseLength;
  }

  public void setQueryLength(int queryLength) {
    this.queryLength = queryLength;
  }

  public int getQueryLength() {
    return queryLength;
  }

  public int getQueryInterval() {
    return queryInterval;
  }

  public void setTransmissionSpeed(long transmissionSpeed) {
    this.transmissionSpeed = transmissionSpeed;
  }

  public long getTransmissionSpeed() {
    return transmissionSpeed;
  }

  public long getBandwidth() {
```

```
        return  bandwidth;
    }


    public int getUserPE() {
        return  userPE;
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        DecimalFormat output2 = new DecimalFormat(" 0.00;-0.00");
        DecimalFormat output4 = new DecimalFormat(" 0.00000000;-0.00000000");
        int numberOfRuns = 1;
        boolean verbose = false;
        double runMeanBandwidth = 0.0; //utilization of the system-side network resource
        double sampleMeanBandwidth = 0.0;
        double runMeanPEUtilization = 0.0; //utilization of the user-side processing resource
        double sampleMeanPEUtilization = 0.0;
        double[] sampleMeansBandwidth = new double[numberOfRuns];
        double sumOfSquaresBandwidth = 0.0;
        double sampleVarianceBandwidth = 0.0;
        double[] sampleMeansPEUtilization = new double[numberOfRuns];
        double sumOfSquaresPEUtilization = 0.0;
        double sampleVariancePEUtilization = 0.0;
        double tInterval95 = 1.984;


        // Here are parameters for the "smart pull" model
        double qtime = 1.0; /*average processing time to generate a query; e.g., 1 second*/
        int qlength = 1000; /*average length of a query in bytes; e.g., 1KB*/
        int qInterval = 600; /*average interval between generating queries in seconds; e.g., 600 (10
minutes)*/
        double rtime = .000001; /*average time in seconds for the information system to respond to
a query; e.g., 1 second*/
        long rlength = 500000; /*average length of a response in bytes; e.g., 500KB*/
        long tspeed = 1250000; /*average transmission speed of the network in bytes/sec; e.g.,
1.25MBps*/
        double ptime = 0.00001; /*average processing time per byte received, in seconds; e.g., 10
microsec*/

        /*Instantiate the SmartPull model*/
        SmartPull sp = new SmartPull(qtime, qlength, qInterval, rtime, rlength, tspeed, ptime);
        if (verbose) {System.out.print(sp);}

        //Define statistics
        SimpleStatsTimeVarying availableBandwidth = new SimpleStatsTimeVarying("bandwidth");
        sp.addPropertyChangeListener(availableBandwidth);
        SimpleStatsTimeVarying PEUtilization = new SimpleStatsTimeVarying("userPE");
        sp.addPropertyChangeListener(PEUtilization);

        //Set for multiple runs to compute mean, standard deviation, and confidence intervals
        int runNumber = 0;
        while (runNumber < numberOfRuns) {
          //Initialize and execute the model
```

```
            simkit.Schedule.stopAtTime(60000.0);
            simkit.Schedule.setVerbose(verbose);
            simkit.Schedule.reset();
            simkit.Schedule.startSimulation();

            //Accumulate statistics across runs
            runMeanBandwidth = 1.0 - availableBandwidth.getMean();
            sampleMeanBandwidth += runMeanBandwidth;
            runMeanPEUtilization = 1.0 - PEUtilization.getMean();
            sampleMeanPEUtilization += runMeanPEUtilization;
            //Store the means for later computation of the sample variance
            sampleMeansBandwidth[runNumber] = runMeanBandwidth;
            sampleMeansPEUtilization[runNumber] = runMeanPEUtilization;

            //Output statistics for each run
            System.out.println("Smart Pull Run Number: " + (runNumber + 1));
            System.out.println("Average Bandwidth Utilization: " +
output4.format(runMeanBandwidth));
            System.out.println("Average PE Utilization: " + output4.format(runMeanPEUtilization));

            runNumber++;
        }

        //compute the sample variance and confidence intervals
        sampleMeanBandwidth = sampleMeanBandwidth / numberOfRuns;
        sampleMeanPEUtilization = sampleMeanPEUtilization / numberOfRuns;
        for (int i=0; i < numberOfRuns; i++) {
            sumOfSquaresBandwidth += (sampleMeansBandwidth[i] -
sampleMeanBandwidth)*(sampleMeansBandwidth[i] - sampleMeanBandwidth);
            sumOfSquaresPEUtilization += (sampleMeansPEUtilization[i] -
sampleMeanPEUtilization)*(sampleMeansPEUtilization[i] - sampleMeanPEUtilization);
        }
        sampleVarianceBandwidth = sumOfSquaresBandwidth / (numberOfRuns - 1);
        sampleVariancePEUtilization = sumOfSquaresPEUtilization / (numberOfRuns - 1);
        double halfIntervalBandwidth = tInterval95 * Math.sqrt(sampleVarianceBandwidth /
numberOfRuns);
        double halfIntervalPEUtilization = tInterval95 * Math.sqrt(sampleVariancePEUtilization /
numberOfRuns);
        System.out.println("******************");
        System.out.println("Smart Pull Results");
        System.out.println("Number of Runs: " + numberOfRuns);
        System.out.println("Sample Mean, Bandwidth Utilization = " +
output4.format(sampleMeanBandwidth));
        System.out.println("95% confidence interval, Bandwidth Utilization = [" +
output4.format((sampleMeanBandwidth - halfIntervalBandwidth)) + ", "
              + output4.format((sampleMeanBandwidth + halfIntervalBandwidth)) + "]");
        System.out.println("Sample Mean, PE Utilization = " +
output4.format(sampleMeanPEUtilization));
        System.out.println("95% confidence interval, PE Utilization = [" +
output4.format((sampleMeanPEUtilization - halfIntervalPEUtilization)) + ", "
              + output4.format((sampleMeanPEUtilization + halfIntervalPEUtilization)) + "]");

    /*
        //Parameters for the "smart push" model
        sp.setQueryTime(2.0); //processing time to generate a query
        sp.setQueryLength(10); //length of a query
```

```
        sp.QueryInterval(0); //interval between generating queries -- no regular
                        //querying, just when re-planning needs to occur
        sp.setResponselength(10); //length of a response
        //transmission speed of the network does not change
        //sp.setResponseInterval(100); //time before valued response

        //Initialize and execute the model
        simkit.Schedule.stopAtTime(60000.0);
        simkit.Schedule.setVerbose(true);
        simkit.Schedule.reset();
        simkit.Schedule.startSimulation();

        //Output statistics for smart push model
        System.out.println("Average Available Bandwidth: " +
output2.format(avgAvailableBandwidth.getMean()));
        System.out.println("Average PE utilization: " + output2.format(1.0 -
avgPEUtilization.getMean()));
   */
        }
}
```

APPENDIX C. SMART PUSH CODE

A. CODE OVERVIEW

The code contained in Appendix C was created by Curtis Blais of the NPS Modeling, Virtual Environments, and Simulation MOVES) Institute for the implementation of the model described in Chapter IV. The code represents the Theory 2 model and will be particularly useful to those desiring an insight into how the assumptions and parameters of the Theory 2 VIRT process as discussed in this thesis were implemented. The code and parameters contained herein may be used as required to further instantiate future simulation models of the VIRT concept.

```
/*
 * SmartPush.java
 *
 * Created on March 9, 2006, 12:00 PM
 */

package virt;

import simkit.*;
import simkit.random.*;
import simkit.stat.*;
import java.util.*;
import java.text.*;
import java.lang.*;

public class SmartPush extends SimEntityBase {

    private double queryTime; //average time to generate a query (conditions of interest), in
seconds
    private RandomVariate queryLength; //random variable for length of a query, in bytes
(conveying conditions of interest)
    private RandomVariate queryInterval; //random variable for time between query initiation, in
seconds
    private RandomVariate responseLength; //random variable for length of the response to a
query, in bytes
    private RandomVariate responseInterval; //random variable for time between generating
responses to user conditions of interest
    private RandomVariate numberOfResponses; //random variable for number of responses to a
user query
    private long transmissionSpeed; //average transmission speed of the network, in bytes/second
    private double processingTime; //average time to process a query response, in seconds

    private int queriesWaiting; //number of queries waiting to be transmitted
    private int responsesWaiting; //number of responses waiting to be processed
```

```
   protected long bandwidth; //network capacity (1 when not in use; 0 when in use) -- for
computing utilization
   protected int userPE; //processing element capacity (1 when not in use; 0 when in use) -- for
computing utilization

   /** Creates a new instance of SmartPull */

   public SmartPush(double queryTime,
           RandomVariate queryLength,
           RandomVariate queryInterval,
           RandomVariate responseInterval,
           RandomVariate numberOfResponses,
           RandomVariate responseLength,
           long transmissionSpeed,
           double processingTime) {

       setQueryTime(queryTime);
       setQueryLength(queryLength);
       setQueryInterval(queryInterval);
       setResponseLength(responseLength);
       setResponseInterval(responseInterval);
       setNumberOfResponses(numberOfResponses);
       setTransmissionSpeed(transmissionSpeed);
       setProcessingTime(processingTime);
   }

   /** Set initial values of all state variables */
   public void reset() {

       super.reset();

       /** StateTransitions for the Run Event */

   }

   public void doRun() {
      userPE = 1;
      firePropertyChange("userPE", getUserPE());

      queriesWaiting = 0;
      responsesWaiting = 0;

      bandwidth = 1; //network resource
      firePropertyChange("bandwidth", getBandwidth());

      waitDelay("InitiateQuery",0.0,new Object[]{},0.0);

   }

   public void doInitiateQuery() {
       /* Code insertion for Event InitiateQuery */

       /* End Code insertion */

       if (userPE > 0) {
           waitDelay("StartQueryProcessing",0.0,new Object[]{},0);
```

```
        }
        else {
            queriesWaiting += 1;
        }
        //schedule the next query generation
        waitDelay("InitiateQuery",getQueryInterval().generate(),new Object[]{},0);
    }

    public void doSendQuery() {
        /* Code insertion for Event SendQuery */

        /* End Code insertion */
        /* StateTransition for bandwidth */
        long _old_Bandwidth = getBandwidth();
        bandwidth = bandwidth - 1;
        firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());

        /* StateTransition for userPE */
        int _old_UserPE = getUserPE();
        userPE = userPE + 1;
        firePropertyChange("userPE", _old_UserPE, getUserPE());

        checkWaiting();

        if (true) {
            double query = getQueryLength().generate();
            waitDelay("NetworkReceiveQuery",query /(double)transmissionSpeed,new Object[]{},0);
        }
    }

    public void doNetworkReceiveQuery() {
        /* Code insertion for Event networkReceiveQuery */

        /* End Code insertion */
        /* StateTransition for bandwidth */
        long _old_Bandwidth = getBandwidth();
        bandwidth = bandwidth + 1;
        firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());

        //schedule query response
        double numberOfResponses = getNumberOfResponses().generate();
        //System.out.println("Responding to query ... " + numberOfResponses + "times");
        waitDelay("NetworkSendResponse",getResponseInterval().generate(),new Object[]{new
Double(numberOfResponses)},0.0);
    }

    public void doNetworkSendResponse(double numberOfResponses) {
        /* Code insertion for Event NetworkSendResponse */

        /* End Code insertion */
        /* StateTransition for bandwidth */
        long _old_Bandwidth = getBandwidth();
        bandwidth = bandwidth - 1;
        firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());
```

```
        if (true) {
            double response = getResponseLength().generate();
            //System.out.println("response message = " + response + " bytes");
            waitDelay("ReceiveResponse",response /(double)transmissionSpeed,new Object[]{new
Double(response)},0);
        }

        if (numberOfResponses > 1) {
            numberOfResponses -= 1;
            //System.out.println("Counting down...numberOfResponses = " + numberOfResponses);
            waitDelay("NetworkSendResponse",getResponseInterval().generate(),new Object[] {new
Double(numberOfResponses)},0.0);
        }
    }

    public void doReceiveResponse(double rLength) {
        /* Code insertion for Event ReceiveResponse */

        /* End Code insertion */
        /* StateTransition for bandwidth */
        long _old_Bandwidth = getBandwidth();
        bandwidth = bandwidth + 1;
        firePropertyChange("bandwidth", _old_Bandwidth, getBandwidth());


        if (userPE > 0) {
            waitDelay("StartProcessingResponse",0.0,new Object[]{new Double(rLength)},0);
        }
        else {
            responsesWaiting += 1;
        }
    }

    public void doStartProcessingResponse(double rLength) {
        /* Code insertion for Event StartProcessingResponse */

        /* End Code insertion */
        /* StateTransition for userPE */
        int _old_UserPE = getUserPE();
        userPE = userPE - 1;
        firePropertyChange("userPE", _old_UserPE, getUserPE());


        if (true) {
            waitDelay("EndResponseProcessing",getProcessingTime() * rLength,new Object[]{},0);
        }
    }

    public void doEndResponseProcessing() {
        /* Code insertion for Event EndResponseProcessing */

        /* End Code insertion */
        /* StateTransition for userPE */
        int _old_UserPE = getUserPE();
        userPE = userPE + 1;
```

```
    checkWaiting();

    firePropertyChange("userPE", _old_UserPE, getUserPE());


}

public void doStartQueryProcessing() {
    /* Code insertion for Event StartQueryProcessing */

    /* End Code insertion */
    /* StateTransition for userPE */
    int _old_UserPE = getUserPE();
    userPE = userPE - 1;
    firePropertyChange("userPE", _old_UserPE, getUserPE());


    if (true) {
        waitDelay("SendQuery",queryTime,new Object[]{},0);
    }
}

public void checkWaiting() {
    if (queriesWaiting > 0) {
        waitDelay("StartQueryProcessing",0.0,new Object[]{},0);
        queriesWaiting -= 1;
    }
    else if (responsesWaiting > 0) {
        waitDelay("StartProcessingResponse",0.0,new Object[]{},0);
        responsesWaiting -= 1;
   }
}

public void setQueryTime(double queryTime) {
    this.queryTime = queryTime;
}

public double getQueryTime() {
    return queryTime;
}

public double getProcessingTime() {
    return processingTime;
}

public void setResponseLength(RandomVariate responseLength) {
    this.responseLength = responseLength;
}

public void setResponseInterval(RandomVariate responseInterval) {
    this.responseInterval = responseInterval;
}

public void setNumberOfResponses(RandomVariate numberOfResponses) {
    this.numberOfResponses = numberOfResponses;
}
```

```
public void setQueryInterval(RandomVariate queryInterval) {
   this.queryInterval = queryInterval;
}

public void setProcessingTime(double processingTime) {
   this.processingTime = processingTime;
}

public RandomVariate getResponseLength() {
   return responseLength;
}

public RandomVariate getResponseInterval() {
   return responseInterval;
}

public RandomVariate getNumberOfResponses() {
   return numberOfResponses;
}

public void setQueryLength(RandomVariate queryLength) {
   this.queryLength = queryLength;
}

public RandomVariate getQueryLength() {
   return queryLength;
}

public RandomVariate getQueryInterval() {
   return queryInterval;
}

public void setTransmissionSpeed(long transmissionSpeed) {
   this.transmissionSpeed = transmissionSpeed;
}

public long getTransmissionSpeed() {
   return transmissionSpeed;
}

public long getBandwidth() {
   return  bandwidth;
}


public int getUserPE() {
   return  userPE;
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
   DecimalFormat output2 = new DecimalFormat(" 0.00;-0.00");
   DecimalFormat output4 = new DecimalFormat(" 0.00000000;-0.00000000");
```

```
	int numberOfRuns = 101;
	boolean verbose = false;

	/*Statistics*/
	double runMeanBandwidth = 0.0; //utilization of the system-side network resource
	double sampleMeanBandwidth = 0.0;
	double runMeanPEUtilization = 0.0; //utilization of the user-side processing resource
	double sampleMeanPEUtilization = 0.0;
	double[] sampleMeansBandwidth = new double[numberOfRuns];
	double sumOfSquaresBandwidth = 0.0;
	double sampleVarianceBandwidth = 0.0;
	double[] sampleMeansPEUtilization = new double[numberOfRuns];
	double sumOfSquaresPEUtilization = 0.0;
	double sampleVariancePEUtilization = 0.0;
	double tInterval95 = 1.984;

	/* Smart Push System Characterization Parameters */
	double qtime = 1.0; /*average processing time to generate a query; e.g., 1 second*/
	long tspeed = 1250000; /*average transmission speed of the network in bytes/sec; e.g.,
1.25MBps*/
	double ptime = 0.00001; /*average processing time per byte received, in seconds; e.g., 10
microsec*/
	//Random Variates -- initial selection of distributions to demonstrate VIRT concepts
	String distribution = "Exponential"; /*for query generation and response generation
intervals*/
	Object[] param = new Object[1];
	param[0] = new Double(6000.0); /*average interval between generating queries in seconds;
e.g., 6000 (100 minutes)*/
	RandomVariate qInterval = RandomVariateFactory.getInstance(distribution, param);
	param[0] = new Double(100); /*average time in seconds between responses to user
conditions of interest*/
	RandomVariate rInterval = RandomVariateFactory.getInstance(distribution, param);
	distribution = "Poisson"; /*for number of responses to a query*/
	param[0] = new Integer(25); /*average number of responses to user conditions of interest*/
	RandomVariate nResponses = RandomVariateFactory.getInstance(distribution, param);
	distribution = "Gamma"; /*for query and response message lengths*/
	Object[] gammaParams = new Object[2];
	gammaParams[1] = new Double(1.0); /*Beta parameter of the Gamma distribution*/
	gammaParams[0] = new Double(1000.0); /*Alpha parameter of the Gamma Distribution*/
	/* note that the mean of a Gamma distribution is Alpha*Beta; variance is Alpha*Beta*Beta */
	/* selected settings describe a distribution with mean equal to Alpha and variance equal to
Alpha */
	/* can adjust the values of Alpha and Beta to create greater variance around a chosen mean
as desired */
	RandomVariate qLength = RandomVariateFactory.getInstance(distribution, gammaParams);
	/*sets the average length of a query in bytes to the value in gammaParams[0]; e.g., 1KB*/
	gammaParams[0] = new Double(1000.0);
	RandomVariate rLength = RandomVariateFactory.getInstance(distribution, gammaParams);
	/*sets the average length of a response in bytes to the value in gammaParams[0]; e.g., 1KB*/

	/*Instantiate the SmartPush model*/
	SmartPush sp = new SmartPush(qtime, qLength, qInterval, rInterval, nResponses, rLength,
tspeed, ptime);
	if (verbose) {System.out.print(sp);}

	//Define statistics
```

```
        SimpleStatsTimeVarying availableBandwidth = new SimpleStatsTimeVarying("bandwidth");
        sp.addPropertyChangeListener(availableBandwidth);
        SimpleStatsTimeVarying PEUtilization = new SimpleStatsTimeVarying("userPE");
        sp.addPropertyChangeListener(PEUtilization);

        //Set for multiple runs to compute mean, standard deviation, and confidence intervals
        int runNumber = 0;
        while (runNumber < numberOfRuns) {
          //Initialize and execute the model
            simkit.Schedule.stopAtTime(6000000.0);
            simkit.Schedule.setVerbose(verbose);
            simkit.Schedule.reset();
            simkit.Schedule.startSimulation();

            //Accumulate statistics across runs
            runMeanBandwidth = 1.0 - availableBandwidth.getMean();
            sampleMeanBandwidth += runMeanBandwidth;
            runMeanPEUtilization = 1.0 - PEUtilization.getMean();
            sampleMeanPEUtilization += runMeanPEUtilization;
            //Store the means for later computation of the sample variance
            sampleMeansBandwidth[runNumber] = runMeanBandwidth;
            sampleMeansPEUtilization[runNumber] = runMeanPEUtilization;

            //Output statistics for each run
            System.out.println("Smart Push Run Number: " + (runNumber + 1));
            System.out.println("Average Bandwidth Utilization: " +
output4.format(runMeanBandwidth));
            System.out.println("Average PE Utilization: " + output4.format(runMeanPEUtilization));

            runNumber++;
        }

        //compute the sample variance and confidence intervals
        sampleMeanBandwidth = sampleMeanBandwidth / numberOfRuns;
        sampleMeanPEUtilization = sampleMeanPEUtilization / numberOfRuns;
        for (int i=0; i < numberOfRuns; i++) {
            sumOfSquaresBandwidth += (sampleMeansBandwidth[i] -
sampleMeanBandwidth)*(sampleMeansBandwidth[i] - sampleMeanBandwidth);
            sumOfSquaresPEUtilization += (sampleMeansPEUtilization[i] -
sampleMeanPEUtilization)*(sampleMeansPEUtilization[i] - sampleMeanPEUtilization);
        }
        sampleVarianceBandwidth = sumOfSquaresBandwidth / (numberOfRuns - 1);
        sampleVariancePEUtilization = sumOfSquaresPEUtilization / (numberOfRuns - 1);
        double halfIntervalBandwidth = tInterval95 * Math.sqrt(sampleVarianceBandwidth /
numberOfRuns);
        double halfIntervalPEUtilization = tInterval95 * Math.sqrt(sampleVariancePEUtilization /
numberOfRuns);
        System.out.println("******************");
        System.out.println("Smart Push Results");
        System.out.println("Number of Runs: " + numberOfRuns);
        System.out.println("Sample Mean, Bandwidth Utilization = " +
output4.format(sampleMeanBandwidth));
        System.out.println("95% confidence interval, Bandwidth Utilization = [" +
output4.format((sampleMeanBandwidth - halfIntervalBandwidth)) + ", "
              + output4.format((sampleMeanBandwidth + halfIntervalBandwidth)) + "]");
```

```
        System.out.println("Sample Mean, PE Utilization = " +
output4.format(sampleMeanPEUtilization));
        System.out.println("95% confidence interval, PE Utilization = [" +
output4.format((sampleMeanPEUtilization - halfIntervalPEUtilization)) + ", "
                + output4.format((sampleMeanPEUtilization + halfIntervalPEUtilization)) + "]");


        }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Mr. Rick Hayes-Roth
   Naval Postgraduate School
   Monterey, California

4. Mr. Curtis Blais
   Naval Postgraduate School
   Monterey, California

5. Mr. John Moschopolous
   PEO IWS6
   Washington Navy Yard

6. Mr. Jim Patzer
   JHU APL
   Laurel, Maryland

7. Mr. Gary Gafke
   JHU APL
   Laurel, Maryland

8. Mr. Dennis Mania
   Air Marine Operations Center (AMOC)
   Riverside, California